

# Índex

<b>1</b>	<b>Introducció</b>	<b>3</b>
1.1	Objectiu	3
1.2	Plantejament del projecte	4
1.3	Planificació	6
<b>2</b>	<b>Marc</b>	<b>10</b>
2.1	El World Wide Web	10
2.1.1	Història d'Internet	10
2.1.2	World Wide Web	11
2.1.3	Funcionament de la Web	13
2.1.4	Implicacions sociològiques de la Web	14
2.2	Estructures client servidor	15
2.2.1	Pàgines estàtiques i dinàmiques	18
2.3	Comunicacions	19
2.3.1	Protocol TCP/IP	19
2.3.2	Protocol HTTP	22
2.3.3	Sockets	24
2.3.4	FTP	26
2.4	Conceptes	26
2.4.1	URL	26
2.4.2	SQL	27
2.4.3	Html	28
2.5	Llenguatge programació	30
2.5.1	PHP	30

<b>2.6</b>	<b>Base de dades</b>	<b>36</b>
2.6.1	Conceptes bàsics	36
2.6.2	MySql	41
<b>2.7</b>	<b>Eines de desenvolupament</b>	<b>42</b>
2.7.1	Adobe Dreamweaver	42
2.7.2	Adobe Fireworks	42
2.7.3	Client ftp Filezilla	43
2.7.4	PhpMyadmin	43
<b>3</b>	<b>Desenvolupament projecte</b>	<b>44</b>
<b>3.1</b>	<b>Definició</b>	<b>44</b>
3.1.1	Parts	45
<b>3.2</b>	<b>Operativa</b>	<b>54</b>
3.2.1	Simulador de dades	55
3.2.2	Adquisició de dades	60
3.2.3	Explotació de dades	61
<b>4</b>	<b>Pressupost</b>	<b>65</b>
<b>4.1</b>	<b>Consideracions prèvies</b>	<b>65</b>
<b>4.2</b>	<b>Costos</b>	<b>65</b>
<b>5</b>	<b>Ampliacions</b>	<b>68</b>
<b>6</b>	<b>Conclusions</b>	<b>70</b>
<b>7</b>	<b>Annexes</b>	<b>72</b>
<b>7.1</b>	<b>Bases de Dades amb phpMyAdmin</b>	<b>72</b>
<b>7.2</b>	<b>Descripció instruccions de sockets</b>	<b>80</b>
<b>7.3</b>	<b>Programació cron job des del servidor</b>	<b>83</b>
<b>8</b>	<b>Bibliografia</b>	<b>88</b>

# 1 Introducció

---

## 1.1 Objectiu

Quan vaig començar a plantejar-me què volia fer en el meu projecte de final de carrera, vaig tenir claríssim que volia que fos relacionat amb la programació. De fet, les assignatures d'informàtica, o en qualsevol que s'hagués de programar, indistintament del llenguatge, són les quines m'ha fascinat més durant la carrera. M'agrada la informàtica i la programació perquè pots veure a la pràctica com funcionen els processos que has programat tu, fent que funcionin com tu has decidit, cercant i provant i tornant a provar quan no fan el què esperes que facin, així com buscant contínuament nous mètodes i solucions que millorin el programa en si.

Per tant, un cop decidit que el projecte havia d'anar per aquest camí, vaig pensar en fer el control d'una maqueta del laboratori a través d'Internet. El tipus i la complexitat de la maqueta no serien l'objectiu del projecte, ja que s'utilitzaria un programa que retornés dades simulades quan l'aplicació creada així ho peticionés, i aquests valors es mostrarien a l'usuari a través de la pàgina web construïda per tal efecte.

Però quan vaig començar a fer el planejament i l'esbós del projecte, i a pensar com hauria de ser el simulador de la maqueta i quins paràmetres hi hauria d'introduir, tant d'entrada com de sortida, vaig veure que seria molt difícil que aquest projecte s'aconguís traslladar a la realitat algun dia, afegint-hi l'abstracció necessària per imaginar-se el funcionament d'una maqueta simulada.

Per tot això, vaig començar a pensar què podia fer, conservant el mateix fons que el projecte plantejat inicialment i que, tot i tractar-se també d'una simulació, fos transportable fàcilment a la realitat. Com que molt sovint miro i llegeixo pàgines web relacionades amb la climatologia i les previsions meteorològiques, se'm va ocórrer que, en comptes d'obtenir dades simulades d'una maqueta del laboratori, les podia obtenir, també simulades, d'estacions meteorològiques suposadament situades en diversos punts del nostre país, aconseguint d'aquesta manera un

projecte amb futur i imaginable per tothom, conservant l'essència i objectiu del plantejament inicial.

Els principals objectius personals d'aquest projecte són, doncs, entendre i aprendre tot el funcionament d'Internet, els protocols i formes de comunicació, així com els diversos llenguatges i plataformes que existeixen per aconseguir crear una pàgina web, sabent valorar al mateix temps totes les avantatges d'utilitzar-ne un o un altre i, d'aquesta manera, fer servir el més adequat al tipus de projecte proposat.

També trobo molt interessant el coneixement de les estructures que han de tenir les bases de dades i tots els conceptes relacionats amb les mateixes, així com els mètodes i llenguatges que permeten inserir i llegir dades de les taules.

## 1.2 Plantejament del projecte

El projecte es tracta, doncs, d'un servei meteorològic anomenat MeteoService UPC que, accessible des d'Internet, proporciona lectures simulades dels paràmetres de 3 "estacions", suposadament instal·lades en tres punts del nostre país i, a més, queda dotat de dades històriques des de la seva posta en marxa. Els simuladors de les estacions es connecten via sockets al servidor http, que va generant la base de dades a mesura que va rebent els valors de les variables meteorològiques. I el client web (usuari), pot consultar, des de qualsevol lloc amb connexió a Internet i en qualsevol moment, aquestes dades meteorològiques a través de la pàgina web preparada i creada per tal efecte.

Aquest projecte es pot separar en tres parts referents al tractament i manipulació de les dades meteorològiques:

- **Simulació de dades.** Cada hora, des de la seva posta en marxa, els 3 emuladors de les estacions del projecte simulen un valor per cada una de les variables meteorològiques (temperatura, humitat relativa i precipitacions) i els envien al servidor, que s'encarregarà d'inserir-los a la base de dades de MeteoService, indexats pel dia i l'hora de la mesura, i deixar-los accessibles per posteriors consultes a través d'Internet. Els valors de les variables

meteorològiques són simulats i generats per rutines que són cridades per les "estacions" quan necessiten dades. Cal dir que la precisió i exactitud dels valors de les variables meteorològiques no és l'objectiu d'aquest projecte.

- **Adquisició** de dades. Les "estacions" exerceixen de clients de sockets amb el servidor d'Internet, i li envien trames amb els valors de les diferents variables meteorològiques. Aquest últim les separa i emmagatzema les dades, indexades per dia i hora de mesura, en la taula corresponent de la base de dades de MeteoService.
- **Explotació de dades.** Pel què fa a les dades, fins ara hem parlat de com es generen els valors meteorològics i com s'insereixen a la base de dades, però cal explicar com s'exploten a nivell d'usuari. Amb aquest objectiu s'ha creat la pàgina web de MeteoService. Qualsevol persona, a través d'Internet i introduint a l'explorador la URL de MeteoService, pot consultar dades meteorològiques, seleccionant prèviament una estació, un dia i una hora concrets.

El servidor d'aquest projecte és un servidor HTTP Apache, que funciona amb Ubuntu Linux. També s'utilitza un servidor ftp per pujar els fitxers i es disposa d'un altre servidor que conté la base de dades composta de les dades històriques de totes les estacions i variables meteorològiques. S'utilitza el Macromedia Dreamweaver per generar les pàgines html, i es combina amb el llenguatge php, que resulta molt pràctic per aplicacions web dinàmiques.

## 1.3 Planificació

A continuació es mostren les diverses fases en les quals s'ha dividit el plantejament, disseny i construcció d'aquest projecte, i les quals hem anat seguint per realitzar-ne el desenvolupament i arribar al resultat final, així com la planificació temporal de mateix exposada en una taula:

### Fase 1. Anàlisi

**Activitat 1.1. ANÀLISI DELS REQUERIMENTS DE L'USUARI** (en aquest cas, del professor tutor i meus)

- Anàlisi de requeriments. Determinació de l'esquema general del projecte, que mostra quins elements formen part del treball, així com la determinació de la comunicació que hi ha d'haver entre cada un d'ells.

**Activitat 1.2. DOCUMENTACIÓ I EINES**

- Adquisició d'informació. Cerca d'informació detallada dels diversos tipus existents de comunicació, llenguatges de programació i estructures de bases de dades, per determinar les més adequades per a la implementació del projecte.
- Estudi de les eines necessàries pel PFC. A partir de la informació obtinguda, determinació de les eines i programes més adequats per cobrir les necessitats i requeriments del treball, així com els requeriments mínims de software per aconseguir l'objectiu
- Cerca a Internet i en manuals d'informació sobre les eines.

**Activitat 1.3. GENERACIÓ DE LA DOCUMENTACIÓ**

- Generació de la documentació relacionada amb les activitats d'aquesta fase per a la memòria del projecte.

### Fase 2. Disseny

**Activitat 2.1. DISSENY DE LA BASE DE DADES METEOSERVICE**

- Disseny de la base de dades. Definició de les diverses taules que componen la base de dades de MeteoService i l'estructura i característiques de cadascuna d'elles.

**Activitat 2.2. DISSENY DE LES FUNCIONS GENERADORES DE VALORS**

- Programa simulador d'estacions meteorològiques. Creació de l'emulador d'estacions meteorològiques que retorna dades dels diversos paràmetres meteorològics per data i hora concrets.
- Funcions simuladores de paràmetres meteorològics. Disseny i programació de les rutines que generen valors aleatoris de les variables meteorològiques de MeteoService. Es crea una funció que retorna Temperatures, una que retorna Humitats Relatives i una que retorna Precipitacions, a partir dels valors dia i hora passats per paràmetre.

**Activitat 2.3. DISSENY DE LA INTERFÍCIE WEB**

- Creació de la pàgina web de MeteoService. Disseny i programació de la pàgina web accessible per usuaris i a través de la qual es realitzen consultes de les dades emmagatzemades en les taules.

**Activitat 2.4. DISSENY DEL MÒDUL DE SOCKETS**

- Estudi i disseny de la comunicació via sockets. Cerca i investigació de totes les instruccions necessàries, tant per part del client com per part del servidor, en una comunicació mitjançant sockets.

**Activitat 2.5. GENERACIÓ DE LA DOCUMENTACIÓ**

- Generació de la documentació relacionada amb activitats d'aquesta fase per a la memòria del projecte.

**Fase 3. Implementació i proves****Activitat 3.1. IMPLEMENTACIÓ DE MÒDULS DE BASES DE DADES**

- Implementació de la base de dades de MeteoService. Inserció de les dades de les estacions a la taula Estacions, així com alguns valors a les taules de les variables meteorològiques, per poder realitzar proves de consulta, mitjançant una rutina per creada per tal finalitat.

**Activitat 3.2. IMPLEMENTACIÓ DE LA PÀGINA WEB**

- Pàgina web de MeteoService. Implementació i validació del funcionament de la pàgina web consultant les dades introduïdes a la base de dades en la fase anterior. Verificació dels requeriments, disseny i funcionament de la web.

**Activitat 3.3. IMPLEMENTACIÓ DE LA COMUNICACIÓ VIA SOCKETS**

- Comunicació sockets client-servidor. Implementació de la comunicació via sockets entre les estacions (client de sockets) i el servidor de sockets i validació i proves de funcionament.
- Inserció de registres a la base de dades. Implementació de la inserció dels valors meteorològics a la base de dades de MeteoService per part del servidor de sockets un cop rebuda la trama per part del client de sockets i separada entre les diverses parts.

**Activitat 3.4. IMPLEMENTACIÓ DE TOTS ELS MÒDULS DE L'APLICACIÓ JUNTS**

- Funcionament de l'esquema dissenyat. A través de la pàgina web, l'usuari pot consultar les dades meteorològiques, escollint l'estació, dia i hora desitjats. L'aplicació accedeix a la base de dades que va construint el servidor a mesura que els programes que emulen les estacions li van enviant dades meteorològiques a través dels sockets.
- Proves de tot el conjunt. Determinació del test amb les validacions de tot el funcionament conjunt a realitzar per verificar que el funcionament és l'adequat segons el disseny i realització del mateix.

**Activitat 3.5. PROGRAMACIÓ DEL SERVIDOR PER DEIXAR FUNCIONANT L'APLICACIÓ DE PERMANENTMENT**

- Execució de les estacions. Des del servidor s'executen els emuladors de les estacions meteorològiques. Cal programar, mitjançant un cron job, el servidor perquè els executi cada hora en punt i, per tant, a cada hora en punt enviïn dades de les variables meteorològiques per emmagatzemar a la base de dades.
- Execució del servidor. Des del servidor http, es programa mitjançant un cron job també, que el servidor de sockets estigui sempre executant-se; és a dir, sempre escoltant i preparat en tot moment per rebre dades de les estacions.

**Activitat 3.6. GENERACIÓ DE LA DOCUMENTACIÓ**

- Generació de la documentació relacionada amb les activitats d'aquesta fase per a la memòria del projecte.



Planificació Temporal																		
Fases/Activitats	Mesos															Hores		
	1	2	3	4	5	6												
<b>Fase 1: Anàlisi</b>																		
<b>Activitat 1.1.</b> ANÀLISI DELS REQUERIMENTS DE L'USUARI																	30	
<b>Activitat 1.2.</b> DOCUMENTACIÓ I EINES																	40	
<b>Activitat 1.3.</b> GENERACIÓ DE LA DOCUMENTACIÓ																	20	
<b>Fase 2. Disseny</b>																		
<b>Activitat 2.1.</b> DISSENY DE LA BASE DE DADES METEOSERVICE																	15	
<b>Activitat 2.2.</b> DISSENY DE LES FUNCIONS GENERADORES DE VALORS																	15	
<b>Activitat 2.3.</b> DISSENY DE LA INTERFÍCIE WEB																	30	
<b>Activitat 2.4.</b> DISSENY DEL MÒDUL DE SOCKETS																	35	
<b>Activitat 2.5.</b> GENERACIÓ DE LA DOCUMENTACIÓ																	25	
<b>Fase 3. Implementació i proves</b>																		
<b>Activitat 3.1.</b> IMPLEMENTACIÓ DE MÒDULS DE BASES DE DADES																	10	
<b>Activitat 3.2.</b> IMPLEMENTACIÓ DE LA PÀGINA WEB																	20	
<b>Activitat 3.3.</b> IMPLEMENTACIÓ DE LA COMUNICACIÓ VIA SOCKETS																	30	
<b>Activitat 3.4.</b> IMPLEMENTACIÓ DE TOTS ELS MÒDULS DE L'APLICACIÓ JUNTS																	50	
<b>Activitat 3.5.</b> PROGRAMACIÓ DEL SERVIDOR PER DEIXAR FUNCIONANT L'APLICACIÓ DE PERMANENTMENT																	25	
<b>Activitat 3.6.</b> GENERACIÓ DE LA DOCUMENTACIÓ																	25	
	TOTAL HORES															<b>370</b>		

# 2 Marc

---

## 2.1 El World Wide Web

### 2.1.1 Història d'Internet

A finals de la dècada del seixanta el Departament de Defensa dels EE.UU va promoure la creació d'una xarxa de comunicació que permetés la interconnexió dels seus proveïdors i els centres d'investigació. A aquest projecte se'l va denominar ARPANET i se'l considera el precursor de l'actual Internet.

Durant la dècada de 1970 havien sorgit diversos protocols per a la seva utilització en ARPANET; no obstant, aquests protocols no eren compatibles, pel què la connexió de xarxes era molt difícil i, per tant, l'intercanvi d'informació molt complicat. Degut a això, ARPA va encarregar el desenvolupament d'un nou protocol denominat Transmission Control Protocol/Internet Protocol (TCP/IP) per la seva utilització en ARPANET. Aquest protocol hauria de permetre la interconnexió de xarxes heterogènies mitjançant l'intercanvi de paquets a través d'encaminadors (routers) o d'adaptadors (gateways). El canvi a aquest protocol es va produir el gener de 1983 i, degut a que el resultat d'aquesta unió va ser una xarxa de xarxes, se la va denominar Internet.

No obstant, Internet seguia sent una xarxa d'utilització complexa, només apta per investigadors i tècnics. La seva interfície era textual i els seus manaments, complexos. La majoria de les operacions es portaven a terme mitjançant el correu electrònic, programes de transferència de fitxers i connexió a sistemes remots. Amb el creixement en la utilització de la xarxa, es fan fer necessàries noves formes per indicar la posició d'un document a recuperar i per navegar a través de la xarxa. Com a resposta a aquestes necessitats, el 1992 va sorgir la Web tal i com la coneixem avui: un fòrum d'intercanvi d'informació i un mercat en creixement accessible a qualsevol usuari independentment del seu grau de coneixement tècnic.

## 2.1.2 World Wide Web

World Wide Web (o la Web) és un sistema de documents d'hipertext i/o d'hipermedis enllaçats i accessibles a través d'Internet. Amb un navegador Web, un usuari visualitza pàgines web que poden contenir text, imatges, vídeos o altres continguts multimèdia, i navegar a través d'elles utilitzant hiperenllaços.

Els orígens del projecte World Wide Web es remunten a l'any 1989 al CERN (Laboratori Europeu de Física de Partícules d'Alta Energia, a Ginebra). Tim Berners-Lee es fa enfrontar al problema de grups d'investigadors geogràficament dispersos que desitjaven accedir als recursos disponibles en punts distants del sistema informàtic del CERN: bases de dades, resultats experimentals, informes de resultats, etc. La seva idea va ser aprofitar les possibilitats que oferien les xarxes d'ordinadors. La interconnexió de recursos permetia accedir a ells des de qualsevol punt de les instal·lacions, i era concebible que s'establissin enllaços (links) entre els recursos per saltar ràpidament d'uns a uns altres.

Per aconseguir aquest propòsit es va dissenyar una arquitectura teòrica complexa, que posava en joc tres estàndards actualment ben coneguts:

- **HTML** (HyperText Markup Language, Llenguatge de Marcat de HiperText), un mètode per codificar la informació dels documents i els seus enllaços.
- **HTTP** (HyperText Transfer Protocol, Protocol de Transferència de HiperText), que especifica com el navegador i el servidor intercanvien informació en forma de peticions i respostes.
- **URL** (Uniform Resource Locator, Localitzador Uniforme de Recursos), que especifica com a cada pàgina d'informació s'associa una adreça única on trobar-la.

Les primeres realitzacions pràctiques es van assajar el 1991 i aviat es va veure que el sistema resultava també idoni a gran escala: tot l'Internet es podia arribar a veure com una xarxa o una teranyina de recursos. Així doncs, el 1992 Tim Berners-Lee va publicar el World Wide Web.

El World Wide Web (WWW o la Web) permet un accés senzill i comprensible a la informació disponible d'Internet, emmagatzemada en forma de pàgines electròniques. A aquestes pàgines se les denomina pàgines web, i poden contenir text, imatge, so i vídeo, podent combinar tots aquests elements dins de les mateixes. Algunes pàgines incorporen animacions o fragments de vídeo, altres afegeixen música de fons. És, per tant, un mètode molt atractiu i complet per

transmetre i presentar la informació.

Un dels principals avantatges que ofereix aquest format per a la presentació d'informació és la utilització dels enllaços o hyperlinks. Un enllaç permet referenciar una pàgina web des d'una altra, enllaçant ambdós continguts. D'aquesta manera, qualsevol pàgina pot incloure referències automàtiques a unes altres pàgines relacionades. Per accedir a la pàgina referenciada, simplement s'ha de seleccionar l'enllaç amb el ratolí i aquesta pàgina es presentarà en pantalla, substituint a l'anterior. Cal destacar que no és necessari que ambdues pàgines es trobin a la mateixa màquina perquè puguin estar enllaçades.

Els **navegadors** (browsers) són programes que permeten accedir a la Web i manipular els documents HTML, tot i que també són capaços de visualitzar altres tipus d'arxiu, tals com imatges (format GIF, JPG, PNG, etc.). Permeten obrir pàgines HTML, de forma local o remota (només caldrà situar l'adreça de la pàgina en la secció corresponent, que s'etiquetarà com a Adreça), contenen un historial de les pàgines més recents a les que s'ha tingut accés i disposen d'una utilitat que permet crear pàgines HTML.

El 1993, al NCSA (National Center for Supercomputing Applications), a la Universitat d'Illinois, es crea el navegador Mosaic, que es faria famós ràpidament. El seu creador, Maarc Andreseen, al poc temps, se'n va anar del NCSA i, juntament amb Jim Clark va fundar Netscape, empresa que va crear Netscape Navigator, que ràpidament va superar en capacitats i velocitat a Mosaic.

Internet Explorer de Microsoft va entrar posteriorment al mercat i avui en dia ha aconseguit desbancar al Netscape Navigator entre els usuaris de Windows. En els últims anys s'ha viscut una autèntica explosió del número de navegadors, que ofereixen cada vegada més integració amb l'entorn de finestres en el que s'executen. Netscape Communications Corporation va alliberar el codi font del seu navegador, naixent així el projecte Mozilla.

A finals del 2004 apareix al mercat Firefox, una rama de desenvolupament de Mozilla que pretén fer-se amb part del mercat d'Internet Explorer. Es tracta d'un navegador més lleuger que el seu germà gran.

Així doncs, actualment el navegador més utilitzat en el món és Internet Explorer, en totes les seves versions. Darrera seu hi ha el navegador Mozilla Firefox, que s'està popularitzant cada vegada més i està arribant a ser un competidor seriós al producte de Microsoft. Existeixen també els navegadors Safari, Netscape Navigator, Opera i Chrome, que tenen una utilització de menys del 2% en el mercat. A més existeix Konqueror del projecte KDE en GNU/Linux i Epiphany, de Gnome.

## 2.1.3 Funcionament de la Web

La visualització d'una pàgina Web de la World Wide Web normalment comença teclejant la URL de la pàgina en el navegador web, o seguint un enllaç de hipertext a aquesta pàgina o recurs. En aquest moment el navegador comença una sèrie de comunicacions, transparents per a l'usuari, per obtenir les dades de la pàgina i visualitzar-la.

El primer pas consisteix en traduir la part del nom del servidor de la URL en una adreça IP utilitzant la base de dades distribuïda a Internet i coneguda com a DNS. Aquesta adreça IP és necessària per contactar amb el servidor web i poder enviar-li paquets de dades.

El següent pas és enviar una petició HTTP al servidor Web sol·licitant el recurs. En el cas d'una pàgina web típica, primer es sol·licita el text HTML i llavors és immediatament analitzat pel navegador, el qual, després, fa peticions addicionals per als gràfics i altres fitxers que formin part de la pàgina.

En rebre els fitxers sol·licitats des del servidor web, el navegador mostra la pàgina tal i com es descriu en el codi HTML, el CSS i altres llenguatges web. Al final s'hi incorporen les imatges i altres recursos per produir la pàgina que veu l'usuari a la seva pantalla.

La majoria de pàgines web contenen hiperenllaços a altres pàgines relacionades i algunes també contenen descàrregues, documents font, definicions i altres recursos web.

Si un usuari accedeix de nou a una pàgina després d'un petit interval, és probable que no es tornin a recuperar les dades del servidor web de la forma en què s'ha explicat anteriorment. Per defecte, els navegadors emmagatzemen en una catxé del disc dur local tots els recursos web als que l'usuari va accedint. El navegador enviarà una petició HTTP només si la pàgina ha estat actualitzada des de la última càrrega, altrament la versió emmagatzemada es reutilitzarà en el pas de "renderitzar" per agilitar la visualització de la pàgina.

Aquest fet és particularment important per reduir la quantitat de tràfic web a Internet. La decisió sobre la caducitat de la pàgina es fa de forma independent per a cada recurs (imatge, full d'estil, fitxers, a més de pel propi codi HTML). No obstant, en llocs de contingut molt dinàmic, molts dels recursos bàsics només s'envien una vegada per sessió.

Apart de les utilitats creades en els servidors Web que poden determinar quan els

fitxers físics han estat actualitzats, els dissenyadors de pàgines web generades dinàmicament poden controlar les capçaleres HTTP enviades als usuaris, de forma que les pàgines intermèdies o sensibles a problemes de seguretat no siguin guardades en catxé. Per exemple, en els bancs on-line i en les pàgines de notícies s'utilitza freqüentment aquest sistema.

## 2.1.4 Implicacions sociològiques de la Web

La Web, tal i com la coneixem avui en dia, ha permès un flux de comunicació global a una escala sense precedents en la història humana. Persones separades en el temps i en l'espai, poden utilitzar la Web per intercanviar –o inclús desenvolupar mútuament- els seus pensament més íntims, o alternativament les seves actituds i desitjos quotidians. Experiències emocionals, idees polítiques, cultura, idiomes musicals, negoci, art, fotografies, literatura... tot pot ésser compartit i disseminat digitalment amb el menor esforç, fent-ho arribar gairebé de forma immediata a qualsevol altre punt del planeta. Tot i que l'existència i la utilització de la Web es basa en tecnologia material, que té a la vegada les seves pròpies desavantatges, aquesta informació no utilitza recursos físics com les biblioteques o la premsa escrita. No obstant, la propagació d'informació a través de la Web (via Internet) no està limitada pel moviment de volums físics, o per còpies manuals o materials d'informació. Gràcies al seu caràcter virtual, la informació en la Web pot ésser buscada més fàcilment i eficientment que en qualsevol altre medi físic, i molt més ràpid del què una persona podria recavar per ella mateixa a través d'un viatge, correu, telèfon o qualsevol altre mitjà de comunicació.

La Web és el major mitjà de difusió d'intercanvi personal aparegut en la Història de la Humanitat, molt per davant de la impremta. Aquesta plataforma ha permès als usuaris interactuar amb molts més grups de persones disperses per tot el planeta del què és possible amb les limitacions del contacte físic, o simplement les limitacions de tots els altres medis de comunicació existents combinats.

Com ja s'ha descrit, l'abast de la Web avui en dia és difícil de quantificar. En total, segons les estimacions de finals del 2007, el número total de pàgines Web, bé d'accés directe mitjançant URL, bé mitjançant l'accés a través de l'enllaç, supera els 155 milions. A la vegada, la difusió del seu contingut és tal, que en poc més de 10 anys, hem codificat mig bilió versions de la nostra història col·lectiva, i la hem

posat a la disposició de 1.000 milions de persones. És, en definitiva, la consecució d'una de les més grans ambicions de l'home: des de l'antiga Mongòlia, passant per la Biblioteca d'Alexandria o la mismíssima Enciclopèdia de Rousseau i Diderot l'home ha intentat recopilar en un mateix temps i lloc tot el saber acumulat des dels seus inicis fins a aquest moment. L'hipertext ha fet possible aquest somni.

## 2.2. Estructures client servidor

Com ja hem anat comentant, la base per entendre com funciona Internet és comprendre que la xarxa no està formada per una única entitat a la que es té accés. Internet és una vasta col·lecció de computadores independents connectades entre si, cada una de les quals proporciona els seus propis serveis. Cada una d'aquestes computadores es denomina host, i totes elles són iguals entre sí des del punt de vista de la capacitat de comunicació, tot i que es distingeixen pel tipus de connexió (mòdem, ethernet, token ring, etc.) i pels serveis que proporcionen.

El protocol HTTP, com molts altres, funciona seguint el model client-servidor. El client (navegador) és una aplicació que executa peticions en el protocol d'aplicació i les envia a través de la xarxa. El servidor roman a l'escolta en un port de la computadora servidor que rep dites peticions, executa el codi necessari per servir-les i respon amb els resultats al client. En aquest model el client és el qui porta la iniciativa en cada sol·licitud, mentre que el servidor es limita a seguir les ordres cursades pel client.

Els servidors, però, són proveïdors de serveis pels clients i porten a terme tasques per ordre dels mateixos. Normalment, un servidor pot donar servei a múltiples clients. Habitualment aquestes computadores són molt potents. Existeixen equips especials per executar servidors d'Internet; no obstant, un simple PC és suficient per instal·lar un servidor d'Internet si el volum de servei requerit no és gaire alt. Normalment, qualsevol sistema es pot configurar com un client, un servidor o com un client-servidor d'Internet.

## Conceptes bàsics del servidor web

Un servidor web és un programa que serveix per respondre les diferents peticions dels navegadors, proporcionant els recursos que es sol·licitin usant el protocol HTTP. Un servidor web bàsic compta amb un esquema de funcionament molt simple, basat en executar infinitament el següent bucle:

1. Espera peticions en el port TCP indicat (l'estàndard per defecte per HTTP és el 80).
2. Rep una petició.
3. Busca el recurs.
4. Envia el recurs utilitzant la mateixa connexió per la que ha rebut la petició.

Un servidor web que segueix l'esquema anterior complirà tots els requisits bàsics dels servidors HTTP, tot i que només podrà fer servir fitxers estàtics.

A partir de l'anterior esquema s'han dissenyat i desenvolupat tots els servidors d'HTTP que existeixen, variant el tipus de peticions que poden atendre. A continuació es detallen algunes de les característiques bàsiques dels servidors web, que amplien òbviament l'esquema anterior.

### Servei de fitxers estàtics

Tots els servidors web han d'incloure, almenys, la capacitat per servir els fitxers estàtics que es trobin en alguna part del disc. Un requisit bàsic és la capacitat d'especificar quina part del disc es servirà. No resulta recomanable que el programa servidor obligui a usar un directori concret, tot i que sí que pot tenir-ne un per defecte.

La majoria de servidors web permeten afegir altres directoris o subdirectoris per servir, especificant en quin punt del "sistema de fitxers" virtual del servidor es localitzaran els recursos.

Alguns servidors web permeten també especificar directrius de seguretat (qui pot accedir als recursos), mentre que altres fan possible l'especificació dels fitxers que s'han de considerar com índex del directori.

### Seguretat i autenticació

La majoria dels servidors web actuals permeten controlar des del programa servidor els aspectes relacionats amb la seguretat i l'autenticació dels usuaris.

### Contingut dinàmic

Un dels aspectes fonamentals del servidor web escollit és el nivell de suport que ofereix per servir contingut dinàmic. La major part del contingut web que se serveix



no ve de pàgines estàtiques sinó que es genera de forma dinàmica, i aquesta tendència va clarament a l'alça; el suport per contingut de tipus dinàmic que ofereix un servidor web és un dels punts crítics de l'elecció.

La major part dels servidors web ofereixen suport per CGI (recordem que els CGI són el mètode més antic i senzill per generar contingut dinàmic). Molts altres ofereixen suport per alguns llenguatges de programació (normalment llenguatges interpretats) com PHP, JSP, ASP, etc. L'oferta és molt àmplia però abans d'escollir un llenguatge de programació de servidor s'ha de plantejar si un desitja un llenguatge molt estàndard per a que l'aplicació no depengui d'un servidor web o d'una arquitectura concreta o si, al contrari, la portabilitat no és prioritària i sí que ho és alguna altre prestació concreta que pugui oferir algun llenguatge de programació concret.

#### Servidors virtuals

Una prestació que guanya acceptació i usuaris ràpidament, molt especialment entre els proveïdors de serveis d'Internet i les empreses d'allotjament de dominis, és la capacitat d'alguns servidors web de facilitar múltiples dominis amb una única adreça IP, discriminant entre els diferents dominis allotjats en funció del nombre de domini enviat en la capçalera HTTP. Aquesta prestació permet l'administració racional i estalviadora d'un bé escàs, les adreces IP. Si es necessiten molts nombres de servidor (perquè proporcionem allotjament o per qualsevol altre motiu) hem d'assegurar-nos que el servidor web escollit ofereixi aquesta facilitat i que el suport per servidors virtuals permeti una configuració diferent per a cada servidor. Seria perfecte que cada servidor es comportés com si fos un ordinador diferent.

#### Prestacions extra

Són moltes les prestacions que ofereixen els diferents servidors web per diferenciar-se de la competència. Algunes són realment útils i poden decidir l'elecció del servidor. S'ha de ser conscient, que si utilitzem algunes d'aquestes característiques, això ens pot lligar a un determinat servidor web i impossibilitar una migració posterior.

## 2.2.1 Pàgines estàtiques i dinàmiques

Com ja explicarem amb més detall, HTML, en realitat, no és un llenguatge de programació sinó, més aviat, es tracta d'un llenguatge descriptiu que té com a objecte donar format al text i a les imatges que pretenem visualitzar al navegador.

A partir d'aquest llenguatge som capaços d'introduir enllaços, seleccionar la mida de les fonts o intercalar imatges, tot això d'una manera prefixada i en cap cas intel·ligent. En efecte, HTML no permet realitzar ni un simple càlcul matemàtic o crear una pàgina a partir d'una base de dades. En realitat, l'HTML, tot i que molt útil a petita escala, resulta bastant limitat a l'hora de concebre grans llocs o portals.

És aquesta deficiència de l'HTML la quina ha fet necessària la utilització d'altres llenguatges molt més versàtils i d'un aprenentatge relativament més complicat, capaços de respondre de manera intel·ligent a les demandes del navegador i que permeten l'automatització de determinades tasques pesades i irremeiables com poden ser les actualitzacions, el tractament de comandes en una botiga virtual, etc.

Aquests llenguatges capaços de recrear a partir de certs "scripts" una infinitat de pàgines automatitzades són els protagonistes del concepte de pàgines dinàmiques.

Així doncs, el contingut d'una pàgina Web pot ser predeterminat, i estaríem parlant d'una **pàgina web estàtica** o generat en el moment de visualitzar-la o sol·licitar-la a un servidor web, i parlariem d'una **pàgina web dinàmica**. Les pàgines dinàmiques que es generen al ser sol·licitades són creades per una aplicació en el servidor web que conté les mateixes.

Cal tenir en compte, però, a l'hora de dissenyar una pàgina web, si estem pensant o volem fer una pàgina web relativament petita, que no necessiti estar al dia contínuament, sinó que els seus continguts són perennes, la utilització de pàgines dinàmiques pot quedar-se gran i pot resultar, inclús, improductiu.

Pel contrari, si el lloc és extens i els seus continguts canvien ràpidament, ens interessa automatitzar, en la mesura del possible, totes les tasques de tal forma que puguem gestionar la seva explotació d'una manera més òptima.

## 2.3 Comunicacions

### 2.3.1 Protocol TCP/IP

La família de protocols d'Internet és un conjunt de protocols de xarxa en els què es basa Internet i que permeten la transmissió de dades entre les xarxes de computadors. Molt sovint se'l denomina conjunt de protocols **TCP/IP**, en referència als dos protocols més importants que el componen: **Protocol de Control de Transmissió (TCP)** i **Protocol d'Internet (IP)**, que foren els dos primers a definir-se, i que són els més utilitzats de la família. Altres protocols d'aquest conjunt són l'HTTP (HyperText Transfer Protocol), l'utilitzat per accedir a les pàgines web, l'FTP (File Transfer Protocol) per a la transferència d'arxius, POP (Post Office Protocol) per correu electrònic o TELNET per accedir a equips remots, entre altres.

El protocol TCP/IP serveix per establir una comunicació entre dos punts remots mitjançant l'enviament d'informació en paquets. Al transmetre un missatge o una pàgina amb imatges, per exemple, el bloc complet de dades es divideix en petits blocs que viatgen d'un punt a un altre de la xarxa, entre dos números IP determinats, seguint qualsevol de les possibles rutes. La informació viatja per molts ordinadors intermitjos a mode de repetidors fins a arribar al seu destí, lloc en el que tots els paquets es reuneixen, es reordenen i es converteixen en la informació original.

La gran avantatge de TCP/IP és que és intel·ligent. Com que cada intercanvi de dades està marcat amb números IP determinats, les comunicacions no tenen perquè creuar-se. I si els paquets no troben una ruta directa, els ordinadors intermitjos proven vies alternatives. Es realitzen comprovacions en cada bloc per tal que la informació arribi intacta, i en el cas que se'n perdi algun, el protocol el sol·licita un altre vegada fins que s'obté la informació completa.

TCP/IP és la base de totes les màquines i software sobre el que funciona Internet: els programes de correu electrònic, transferència d'arxius i transmissió de pàgines amb text i imatges i enllaços de hipertext.

La família de protocols d'Internet pot descriure's per analogia amb el model OSI (Open System Interconnection), que descriu els nivells o capes de la pila de protocols, tot i que a la pràctica no correspon exactament amb el model d'Internet.

El model d'Internet va ser dissenyat com la solució a un problema pràctic d'enginyeria. El model OSI, en canvi, va ser proposat com una aproximació teòrica i també com una primera fase en l'evolució de les xarxes d'ordinadors. Per tant, el model OSI és més fàcil d'entendre, però el model TCP/IP és el quin realment s'utilitza.

En una pila de protocols, cada nivell soluciona una sèries de problemes relacionats amb la transmissió de dades, i proporciona un servei ben definit als nivells més alts. Els nivells superiors són els més propers a l'usuari i tracten amb dades més abstractes, deixant als nivells més baixos la tasca de traduir les dades de forma que siguin físicament manipulables.

Així doncs, el protocol TCP/IP és una arquitectura de diversos nivells en la que els protocols de cada nivell tenen una comesa específica sobre la que es recolzen els protocols de nivells superiors.

5	Capa d'Aplicació	Ex. HTTP, DNS, FTP
4	Capa de Transport	Ex. TCP, UDP, RTP, SCTP,
3	Capa d'Internet	Per TCP/IP aquest és el Protocol d'Internet
2	Capa d'Accés a la xarxa	Ex. Ethernet, Token Ring, HDLC, RDSI
1	Capa Física	Ex. mitjà físic, i tècniques de codificació, T1, E1

### Capa Física

La **capa física** defineix la interfície física de la comunicació, entre el dispositiu de transmissió de dades (per exemple, l'estació de treball o la computadora) i el mitjà de transmissió o xarxa. Aquesta capa s'encarrega de l'especificació de les característiques del medi de transmissió, la naturalesa dels senyals, la velocitat de dades i qüestions afins.

### Capa d'accés a la xarxa

La **capa d'accés a la xarxa** és responsable de l'intercanvi de dades entre el sistema final (servidor, estació de treball, etc.) i la xarxa a la qual està connectat. L'emissor ha de proporcionar a la xarxa l'adreça del destí, de tal manera que

aquesta pot encaminar les dades fins al destí apropiat. L'emissor pot requerir certs serveis que poden ser proporcionats pel nivell de xarxa, per exemple, sol·licitar una determinada prioritat. El software en particular que es faci servir en aquesta capa dependrà del tipus de xarxa que es disposi. Així, s'han desenvolupat, entre altres, diversos estàndards per la commutació de circuits, la commutació de paquets (per exemple, retransmissió de trames) i per les xarxes d'àrea local (per exemple, Ethernet). Per tant, té sentit separar en una capa diferent totes aquelles funcions que tinguin a veure amb l'accés a la xarxa. Fent això, el software de comunicacions situat per sobre de la capa d'accés a la xarxa no haurà d'ocupar-se dels detalls específics de la xarxa a utilitzar. El software de les capes superiors hauria de, per tant, funcionar correctament amb independència de la xarxa a la que el commutador està connectat.

### Capa Internet

Per sistemes finals connectats a la mateixa xarxa, la capa d'accés a la xarxa està relacionada amb l'accés i encaminament de les dades. En situacions en les que els dos dispositius estiguin connectats a xarxes diferents es necessitaran una sèrie de procediments que permetin que les dades travessin les diferents xarxes interconnectades. Aquesta és la funció de la **capa Internet**. El protocol Internet (IP, Internet Protocol) s'utilitza en aquesta capa per oferir el servei d'encaminament a través de varies xarxes. Aquest protocol s'implementa tant en els sistemes finals com en els encaminadors intermedis. Un encaminador és un processador que connecta dues xarxes i la funció principal del qual és retransmetre dades des d'una xarxa a una altra seguint la ruta adequada per arribar al destí.

### Capa extrem-a-extrem o capa de transport

Independentment de la naturalesa de les aplicacions que estiguin intercanviant dades, és usual requerir que les dades s'intercanviïn de forma fiable. És a dir, seria desitjable assegurar que totes les dades arribin a l'aplicació destí, en el mateix ordre en que van ser enviades, sense duplicats ni pèrdues, i lliures d'errors. Els mecanismes que proporcionen aquesta fiabilitat són essencialment independents de la naturalesa intrínseca de les aplicacions. Per tant, té sentit agrupar tots aquests mecanismes en una capa comú compartida per totes les aplicacions; aquesta es denomina **capa extrem-a-extrem o capa de transport**. El protocol per al control de la transmissió, TCP (Transmission Control Protocol), és el més utilitzat per proporcionar aquesta funcionalitat.

### Capa d'aplicació

Finalment, la **capa d'aplicació**, conté tota la lògica necessària per possibilitar les diferents aplicacions d'usuari. Per a cada tipus particular d'aplicació, com per exemple la transferència d'arxius, es necessitarà un mòdul ben diferenciat. En aquest nivell se situa el protocol HTTP (HyperText Transfer Protocol) que és l'utilitzat en la programació d'aplicacions web.

## 2.3.2 Protocol HTTP

El protocol de Transferència de HyperText (HyperText Transfer Protocol, HTTP) és un protocol client-servidor que defineix la sintaxis i la semàntica de l'intercanvi d'informació entre els clients i els servidors de la Web i, més concretament, entre els navegadors (com Internet Explorer, Netscape Navigator o Mozilla Firefox) i els servidors HTTP (com Apache o Internet Information Server). És un protocol d'aplicació que funciona sobre els serveis de TCP/IP seguint el mateix esquema que altres serveis de xarxa. En cada màquina amb servei HTTP, hi ha un servidor escoltant en un port (habitualment el 80) i esperant les peticions de servei dels clients web. Una vegada s'estableix la connexió, el protocol TCP s'encarrega de mantenir la comunicació i garantir un intercanvi de dades lliure d'errors.

HTTP utilitza un esquema client-servidor amb missatges autocontinguts per facilitar l'intercanvi de dades de les operacions. Els clients es connecten al servidor i li envien missatges al seu port 80 utilitzant TCP/IP. Aquests missatges han de tenir un format ben definit en el protocol HTTP, essent fonamental el manament adjunt als mateixos. Després de processar la informació, el servidor respon amb un altre missatge que pot incloure estats, dades o codis d'error, segons el resultat de l'operació sol·licitada. Totes les operacions poden adjuntar un objecte o recurs sobre el que actuen; cada objecte web (document HTML, arxiu multimèdia o aplicació CGI) és conegut per la seva URL, i el servidor i els clients saben què fer amb ells perquè utilitzen descripcions MIME (Multipurpose Internet Mail Extensions), que descriuen el tipus de document.

El protocol HTTP utilitza els manaments bàsics exposats a la següent taula:

<b>Manament</b>	<b>Descripció</b>
GET	Sol·licita llegir una pàgina web
HEAD	Sol·licita llegir la capçalera d'una pàgina web
PUT	Sol·licita emmagatzemar una pàgina web
POST	Envia dades a una aplicació web
DELETE	Esborra la pàgina web
LINK	Connecta a dos recursos existents
UNLINK	Trenca una connexió existent entre dos recursos

Aquests manaments es poden utilitzar també per manipular qualsevol tipus d'arxius, encara que no siguin pàgines web, si bé en aquest cas, els clients han de saber amb quin tipus d'objectes tracten o tenir una descripció MIME dels mateixos.

Per adaptar bé el protocol a l'intercanvi d'informació a Internet i dotar-la de major senzillesa, el protocol HTTP té, a més, altres característiques: comunicació amb caràcters de 8 bits (d'aquesta manera, es pot transmetre qualsevol tipus de document: text, binari, etc., respectant el seu format original), no té estat i genera una connexió per cada operació. Com que no guarda cap informació sobre connexions anteriors i el desenvolupament d'aplicacions web necessita freqüentment mantenir un estat, s'utilitzen "cookies", que és informació que un servidor pot emmagatzemar en el sistema client. Això permet a les aplicacions web instituir la noció de "sessió" i també permet rastrejar usuaris, donat que el client manté guardats els cookies per un temps indeterminat.

## 2.3.3 Sockets

El concepte de socket, així com la programació dels sockets, van ser desenvolupats durant els anys 80 en l'entorn de comunicació Unix a la universitat de Berkeley.

Socket designa un concepte abstracte pel qual dos programes poden intercanviar-se qualsevol fluxe de dades, generalment de forma ordenada i fiable.

Així doncs, la utilització de sockets proporciona un eina que habilita la comunicació i permet l'intercanvi de dades entre dos programes que poden o no estar situats en diferents ordinadors. Un socket pot considerar-se com un punt final en una comunicació. Per tal que aquesta comunicació sigui possible, el socket, compta amb tres recursos que el defineixen: una adreça IP, un número de port i un protocol. Aquests atributs permeten el reconeixement de la computadora, la identificació del programa dins d'ella i l'intercanvi de dades, respectivament i es defineixen a continuació:

- **Adreça IP.** És un número que identifica de manera lògica i jeràrquica a una interfície d'un dispositiu (habitualment una computadora) dins una xarxa que utilitzi el protocol IP.
- **Número de port.** És una forma genèrica de denominar a una interfície per la qual diferents tipus de dades poden ser enviades i rebudes.
- **Protocol.** Són el conjunt de normes i procediments útils per a la transmissió de dades, coneguts per l'emissor i el receptor.

Els sockets permeten implementar una **arquitectura client-servidor**. La comunicació ha de ser iniciada per un dels programes que es denomina programa *client*. El segon programa espera a que l'altre iniciï la comunicació, per aquest motiu es denomina programa *servidor*.

Un socket és un fitxer existent en la màquina client i en la màquina servidora, que serveix en última instància perquè el programa servidor i el client llegeixin i escriguin la informació. Aquesta informació serà transmesa per les diferents capes de xarxa.

Els sockets dependran del protocol de comunicacions en el qual estiguin basats. Generalment el protocol que més s'utilitza és el denominat a partir de les inicials TCP (Transmission Control Protocol), tot i que no és l'únic possible. Aquest permet, per exemple, establir una relació servidor-client i habilitar el traspàs d'informació entre ells de forma ordenada.



El servidor permanece passiu a l'espera que el client li sol·liciti connexió. Per establir aquesta connexió són necessaris els tres conceptes centrals que ja hem mencionat anteriorment. L'adreça IP necessària és la del servidor no la dels clients, donat que són ells els que sol·liciten la connexió amb el servidor i no viceversa.

L'avantatge en la utilització de sockets radica en que aquests recursos permeten la transferència de les dades sense errors o omissió i en el mateix ordre en el que han estat transmesos.

Existeixen bàsicament dos tipus de "canals de comunicació" o sockets, els orientats a connexió i els no orientats a connexió.

- **Orientats a connexió.** Ambdós programes han de connectar-se entre ells amb un socket i fins que no estigui establerta correctament la connexió, cap dels dos pot transmetre dades. Aquesta és la part TCP del protocol TCP/IP, i garanteix que totes les dades arribaran a l'altre programa correctament. S'utilitza quan la informació a transmetre és important, quan no es pot perdre cap dada i no importa que els programes puguin quedar "bloquejats" esperant o transmetent dades.
- **No orientats a connexió.** En aquest cas no és necessari que els programes es connectin. Qualsevol d'ells pot transmetre dades en qualsevol moment, independentment que l'altre programa estigui "escoltant" o no. És l'anomenat protocol UDP, i garanteix que les dades que arribin siguin correctes, però no garanteix que arribin tots. S'utilitza quan és molt important que el programa no es quedi bloquejat i no importa que es perdin dades.

## 2.3.4 FTP

**El protocol de transferència d'arxius (FTP, File Transfer Protocol)** s'utilitza per enviar arxius d'un sistema a un altre sota el control de l'usuari. Es permet transmetre arxius tant de text com en binari. A més a més, el protocol permet controlar l'accés dels usuaris. Quan un usuari sol·licita la transferència d'un arxiu, FTP estableix una connexió TCP amb el sistema destí per intercanviar missatges de control. Aquesta connexió permet a l'usuari transmetre el seu identificador i contrasenya, a més de la identificació de l'arxiu juntament amb les accions a realitzar sobre el mateix. Una vegada que l'arxiu s'hagi especificat i la seva transferència hagi estat acceptada, s'establirà una segona connexió TCP a través de la qual es materialitzarà la transferència. L'arxiu es transmet a través de la segona connexió, sense necessitat d'enviar informació extra o capçaleres generades per la capa d'aplicació. Quan la transferència finalitza, s'utilitza la connexió de control per indicar la finalització. A més, aquesta mateixa connexió estarà disponible per acceptar noves ordres de transferència.

## 2.4 Conceptes

### 2.4.1 URL

URL significa Uniform Resource Locator, és a dir, localitzador uniforme de recurs. És una seqüència de caràcters, d'acord a un format estàndard, que s'utilitza per nombrar recursos, com documents i imatges a Internet, per a la seva localització.

Els URL van ser una innovació fonamental en la història d'Internet. Van ser utilitzades per primera vegada pel Tim Berners-Lee el 1991, per permetre als autors de documents establir hiperenllaços a la World Wide Web (WWW o Web). Des de 1994, en els estàndards d'Internet, el concepte de URL ha estat incorporat en el més general de URI (Uniform Resource Identifier – Identificador Uniforme de Recurs), però el terme URL encara s'utilitza àmpliament.

L'URL és la cadena de caràcters amb la qual s'assigna una adreça única a cada un

dels recursos d'informació disponibles a Internet. Existeix un URL únic per cada pàgina de cada un dels documents de World Wide Web.

L'URL d'un recurs d'informació és la seva adreça a Internet, la qual permet que el navegador la trobi i la mostri de forma adequada. Per això l'URL permet especificar el protocol que s'utilitza per obtenir el document, el nom del host que el conté, el port TCP/IP en el què està escoltant el servidor que conté el document i el document en sí.

El format general d'un URL és:

**protocol://màquina/directori/arxiu**

Per exemple: `http://juncadella.net/meteoservice/public/web1.php`

També poden afegir-se altres dades:

`Protocol://usuari:contrasenya@màquina:port/directori/arxiu`

Tot i que, normalment, podem observar que en la majoria d'URL's no s'especifica el port del servidor al que s'ha de connectar. La raó és que, per cada servei, existeix un port predefinit i, si no s'especifica, s'utilitza aquest número de port. Com ja hem comentat anteriorment, en el cas del protocol HTTP el port predefinit és el 80.

## 2.4.2 SQL

El llenguatge de consulta estructurat (SQL, Structured Query Language) és un llenguatge declaratiu d'accés a bases de dades relacionals que permet especificar diversos tipus d'operacions sobre les mateixes. Una de les seves característiques és el maneig de l'àlgebra i el càlcul relacional permetent llançar consultes amb el fi de recuperar –d'una forma senzilla– informació d'interès d'una base de dades, així com també fer canvis sobre la mateixa. És un llenguatge de quarta generació (4GL), un llenguatge de programació en el que s'especifica quins resultats es volen obtenir, a diferència dels 3GL, on s'especifica com han d'obtenir-se els resultats.

## Orígens i evolució

Els orígens de l'SQL estan lligats als de les bases de dades relacionals. Quan, el 1970 E. F. Codd va proposar el model relacional, també va associar-hi un subllenguatge d'accés a les dades basat en el càlcul de predicats. Basant-se en aquestes idees, els laboratoris d'IBM defineixen el llenguatge SEQUEL (Structured English QUery Language), que més tard seria àmpliament implementat pel SGBD experimental System R, desenvolupat el 1977 també per IBM. No obstant, va ser Oracle qui va introduir-lo per primera vegada el 1977 en un programa comercial.

El SEQUEL acabaria sent el predecessor de SQL, sent aquest una versió evolucionada del primer. L'SQL passa a ser el llenguatge per excel·lència dels diversos SGBD relacionals sorgits en els anys següents i el 1986 ANSI l'estandaritza finalment, donat lloc a la primera versió estàndard d'aquest llenguatge, l'"SQL-86" o "SQL1". L'any següent aquest estàndard és també adoptat per l'ISO.

Però el primer estàndard no cobreix totes les necessitats dels desenvolupadors i inclou funcionalitats de definició d'emmagatzemament que es van considerar suprimir. Així que el 1992 es llança un nou estàndard ampliat i revisat de l'SQL anomenat "SQL-92" i "SQL2".

En l'actualitat, l'SQL és l'estàndard de la immensa majoria dels SGBD comercials.

## 2.4.3 Html

Les pàgines HTML (Hyper Text Markup Language) no són més que text, al que se li han afegit certes etiquetes per tal que qualsevol navegador sàpiga què fer per presentar la informació en pantalla. Aquestes etiquetes tenen atributs que li indiquen com fer aquesta tasca.

Per tant, un document HTML es pot llegir i modificar utilitzant un editor de text. Al llarg del text s'aniran veient algunes etiquetes, amb els seus atributs, però pel demès és codi llegible. Llavors, cada cop que un usuari es connecti a un website, aquest lloc li enviarà al navegador un arxiu HTML que conté la pàgina sol·licitada. El navegador, al rebre aquest arxiu, el llegeix i l'interpreta. Les etiquetes li indiquen com representar el text, amb quina font, color o format, i on ha de situar les taules o imatges, construint així la informació que posteriorment es veurà en pantalla.

## La pàgina HTML mínima

Tot document HTML comença amb `<HTML>` i acaba amb `</HTML>`. Cada un d'aquests elements s'anomena etiqueta i, normalment, hi ha una etiqueta d'inici i una altre de final, que marquen la regió o àrea que es veu afectada per la mateixa etiqueta. Aquestes etiquetes es diferencien en que la de final conté el caràcter `/` al davant.

Les etiquetes, per diferenciar-les del text, van entre símbols de menor `"<"` i major `">"`, que indiquen on comença i acaba l'etiqueta. El que hi ha entre aquests dos símbols és l'etiqueta pròpiament dita, que no es veurà en pantalla, ja que la seva funció és indicar al navegador com s'ha de representar la informació en pantalla.

Tota pàgina HTML ha de tenir una capçalera, que ve delimitada per l'etiqueta `<HEAD>` i el seu tancament `</HEAD>`, i un cos, indicat amb l'etiqueta `<BODY>` i el seu tancament `</BODY>`.

```
<HTML>

<HEAD>

<TITLE> MeteoService UPC</TITLE>

</HEAD>

<BODY>

Aquesta és la pàgina de MeteoService UPC

</BODY>

</HTML>
```

## Formularis

Els formularis en HTML tenen la mateixa utilitat que en la vida real, permeten recaptar informació dels usuaris, les seves dades personals, la seva opinió, etc.

La principal diferència amb les altres pàgines HTML és que afegixen interactivitat amb l'usuari donat que les coses evolucionaran de forma diferent en funció de les dades que hagi introduït.

Un formulari es construeix a partir de dues etiquetes bàsiques:

- `<FORM>` I `</FORM>`. Marquen l'inici i el fi d'un formulari.
- `<INPUT>`. Sol·licita informació a l'usuari, és una entrada del formulari. No té etiqueta de tancament. Pot ser de molts tipus, i n'hi solen haver varies entre el començament i el final d'un formulari.

## 2.5 Llenguatge programació

### 2.5.1 PHP

#### Introducció

PHP són les inicials de PHP: HyperText Preprocessor i es tracta d'un llenguatge interpretat que s'executa al servidor i que permet la creació de documents i pàgines web amb contingut dinàmic.

Les principals característiques del llenguatge són:

- És un llenguatge de programació interpretat que s'executa al servidor i que permet la creació de documents HTML dinàmics. Cal destacar que PHP és de lliure distribució i que es pot executar en entorns Unix/Linux.
- Presenta una sintaxis similar a la de C, Perl, Java i JavaScript, que facilita el seu aprenentatge.
- Permet la connexió i l'accés a multitud de sistemes de gestió de bases de dades. També disposa de mecanismes per al processament d'arxius, tractament de text, generació dinàmica de pàgines i tractament de documents XML.
- Ofereix suport a múltiples protocols de comunicacions d'Internet, per exemple, HTTP, IMAP, FTP, etc.
- El seu codi font és gratuït i obert. A més, es troba disponible per moltes plataformes, característica que ofereix l'avantatge que les aplicacions desenvolupades en PHP es puguin transportar d'una plataforma a una altra sense necessitat de modificar ni una línia de codi.

#### Invocació

El codi PHP està embotrat dins d'un document HTML. Quan un usuari des del seu navegador fa clic sobre un enllaç corresponent a un document HTML que inclou codi PHP, el navegador realitza una sol·licitud al servidor web corresponent. El servidor localitza el document, detecta que es tracta d'una pàgina amb codi PHP i posa en funcionament l'interpret del llenguatge. L'interpret executa el codi PHP i genera un resultat, generalment una pàgina web, que es retorna al navegador per tal que realitzi la seva visualització.

Hi ha diverses maneres d'introduir codi PHP en un document HTML. Les més habituals són:

- Insertant el codi PHP entre les etiquetes `<?php i ?>`, o simplement entre `<? i ?>`.
- Amb l'etiqueta `<SCRIPT>` de HTML. El codi PHP s'ha d'insertar entre les etiquetes `<SCRIPT LANGUAGE="php">` i `</SCRIPT>`.

El següent fragment de codi mostra una primera pàgina amb codi PHP:

```
<HTML>
<HEAD>
    <TITLE> Primer programa amb PHP </TITLE>
</HEAD>

<BODY>
    <H1>
    <CENTER>
    <?
        Echo "Primer exemple en PHP";
    ?>
    </CENTER>
    </H1>
</BODY>
</HTML>
```

## Descripció del llenguatge

### Comentaris

1. Entre `/* i */`. Poden ocupar diverses línies.
2. Comença amb `//` i finalitza al final de la línia.
3. Comença amb `#` i finalitza al final de la línia.

### Tipus i variables

Una variable és un nom simbòlic que s'utilitza per emmagatzemar els valors amb els que es vol operar dins del programa.

El nom d'una variable és una cadena alfanumèrica que identifica una zona de memòria on s'emmagatzema el seu valor. Hi ha diverses limitacions pel que fa al nom que es pot assignar a una variable:

- Es compon d'una sèrie de caràcters que poden ser una lletra ([A-Z] i [a-z]), excloent les lletres accentuades i la ñ, un dígit o \_.
- La variable ha de començar amb el símbol \$, al que seguirà una cadena de caràcters que començarà una lletra o el caràcter de subratllat.
- PHP és sensible a les majúscules i les minúscules; per tant, \$numero i \$Numero serien dues variables diferents.

Una característica important relacionada amb les variables és que aquestes no necessiten definir-se abans d'utilitzar-se i poden contenir qualsevol valor; és a dir, en un moment determinat poden emmagatzemar un número i en un altre una cadena de caràcters.

L'assignació d'un valor a una variable es realitza mitjançant l'operador =.

En PHP existeixen 4 tipus de dades bàsiques:

<b>Número enter</b>	Número enter, positiu o negatiu
<b>Número en coma flotant</b>	Números amb notació decimal o científica
<b>Booleà</b>	Valor de tipus booleà
<b>Cadena de caràcters</b>	Cadenes de caràcters tancades entre cometes simples o dobles

PHP permet el que s'anomena com variables de variables. Així, en la següent assignació:

```
$color = "verd"; (variable anomenada $color amb valor "verd")
$$color = "poma"; (assigna el valor "poma" a la variable el nom de la qual és el
contingut de la variable $color). Aquesta assignació és similar a:
$verd = "poma";
```

D'aquesta forma, es pot fer referència a les variables \$color, \$\$color i \$verd, les quals tindran els valors "verd", "poma" i "poma" respectivament.

### Conversió de tipus

PHP permet realitzar la conversió d'un tipus a un altre mitjançant el que popularment s'anomena *casting*. Aquest *casting* o conversió explícita de tipus es porta a terme amb la següent sintaxis:

```
(tipus de dades) expressió;
```



Amb aquesta sentència forcem la conversió d'expressió al tipus de dades que s'indica entre parèntesis. Els tipus que es poden substituir són: `int` o `long` per enters, `real`, `double` o `float` per números en coma flotant, `string` per cadenes de caràcters, `Boolean` o `bool` per tipus de dades booleans, `array` per arrays i `object` per objectes.

Com que PHP no és un llenguatge amb molts tipus diferents, hi poden haver ocasions en què sigui necessari conèixer el tipus de dades que emmagatzema una determinada variable. Podem utilitzar les següents funcions per fer-ho:

<b>Funció</b>	<b>Descripció</b>
<code>is_int(var)</code>	Torna <code>true</code> si <code>var</code> és de tipus enter
<code>is_long(var)</code>	Torna <code>true</code> si <code>var</code> és de tipus enter
<code>is_double(var)</code>	Torna <code>true</code> si <code>var</code> és de tipus real
<code>is_float(var)</code>	Torna <code>true</code> si <code>var</code> és de tipus real
<code>is_real(var)</code>	Torna <code>true</code> si <code>var</code> és de tipus real
<code>is_string(var)</code>	Torna <code>true</code> si <code>var</code> és una cadena
<code>is_numeric(var)</code>	Torna <code>true</code> si <code>var</code> és de tipus numèric
<code>is_array(var)</code>	Torna <code>true</code> si <code>var</code> és un array
<code>is_object(var)</code>	Torna <code>true</code> si <code>var</code> és un objecte

### Constants

PHP permet la definició de constants, que permeten utilitzar identificadors que no canvien de valor durant l'execució d'un programa. La següent sentència:

```
define(PI, 3.1416)
```

defineix la constant `PI` amb el valor real 3.1416.

### Sentències de control

- **Funcions.** Una funció és un element d'un programa que realitza una determinada tasca. La utilització de funcions permet dividir un programa en components més petits i senzills: les funcions.

El format general de definició d'una funció és el següent:

```
function NomFuncio($parametre1, $parametre2, ...)
{
    // COS DE LA FUNCIO
}
```

Els arguments que es passen a una funció poden ser variables, números o objectes. Una funció també pot retornar un resultat utilitzant la sentència `return`.

#### Pas de paràmetres a una funció

- **Pas de paràmetres per valor.** És el mètode utilitzat per defecte en PHP. Quan una variable es passa per valor, la funció no pot modificar aquesta variable dins de la funció. Encara que la funció realitzi modificacions sobre la mateixa, els seus efectes no es veuran fora de la funció.
- **Pas de paràmetres per referència.** En aquest cas, la funció pot modificar el valor de la variable i els canvis que es produeixin en la mateixa seran visibles fora de la funció. Per indicar en PHP el pas d'una variable per referència s'utilitza el caràcter `&` en la declaració que es faci de la mateixa en la funció.

#### Àmbit i tipus de variables

L'àmbit d'una variable defineix la visibilitat de la mateixa, és a dir, des d'on es pot accedir a aquesta variable. Així, les variables podran ser locals o globals.

Una *variable local* és aquella que es defineix dins d'una funció. El seu àmbit es restringeix a la funció en la que es defineix i, per tant, no és visible des de fora de la mateixa.

Una *variable global* és la que es defineix fora de les funcions i, per tant, pot ser accedida des de qualsevol funció, però per això ha d'estar declarada en la funció utilitzant la paraula `global`. Una altra forma d'accedir a variables globals es realitza mitjançant l'array `$GLOBALS` que emmagatzema totes les variables globals que hagin estat definides en tot el codi PHP.

PHP també permet la definició de variables estàtiques. Una *variable estàtica* és una variable local a una funció el valor de la qual persisteix entre les diferents invocacions a la funció.

## Arrays

Particularitats dels arrays en PHP:

- L'accés als arrays es pot fer de forma indexada. En aquest cas, l'índex del primer element és 0.
- Els elements d'un array poden ser de diferent tipus; per exemple:

```
$vector[0] = 84;
$vector[1]=12.98;
$vector[2]="numero";
```

- Un array pot tenir índexos no seqüencials.

```
$vector[40] = 84;
$vector[10]=12.98;
$vector[1]="numero";
```

Per saber el número d'elements d'un array es pot utilitzar la funció `count()`. Per exemple:

```
print(count($vector));
```

- Un array també es pot crear amb un constructor `array`. Per exemple:

```
$alumnes = array("Pere", "Joan", "Anna", "Laura"); //array de 4
elements, el primer corresponent a l'element amb índex 0
```

- Es permet la definició i utilització d'arrays associatius. Un array associatiu està format per parells clau-valor i l'accés als elements es realitza utilitzant una determinada clau. Per exemple (conté dades relatives a un alumne):

```
$Alumne["Nom"]="Carles";
$Alumne["Cognoms"]="Sánchez";
$Alumne["Edat"]="20";
$Alumne["Matricula"]="345678";
```

## Recorregut d'arrays associatius

Un aspecte important a tenir en compte amb els arrays associatius és que els diferents elements que componen un array no mantenen cap ordre entre ells, cosa que impedeix accedir als elements d'un array per la seva posició. Quan es vol recórrer de forma seqüencial els elements d'un array associatiu, es pot utilitzar el bucle `foreach` o la funció `each()`.

Manipulació de dates i hores➤ **Funció `getdate()`**

La funció `getdate()` obté un array associatiu amb la informació relativa a l'hora i la data actual del sistema.

Inclusió de codi des d'un arxiu

PHP permet executar codi existent en altres arxius. Hi ha dos funcions que permeten realitzar la inclusió del codi existent en arxius externs: `require()` i `include()`.

## 2.6 Base de dades

### 2.6.1 Conceptes bàsics

A continuació es defineixen els conceptes bàsics pel què fa a bases de dades:

- **Base de dades.** Conjunt de dades pertanyents a un mateix concepte i emmagatzemats sistemàticament per la seva utilització posterior, i que llur contingut és coherent a la realitat que representa.
- **Model de dades.** És un conjunt de conceptes i regles que permeten estructurar les dades resultants de l'observació de la realitat de manera que quedin representades les propietats, tant estàtiques com dinàmiques, d'ella.
- **Model Relacional.** El Model Relacional de dades va ser proposat per E. Codd el 1970 imposant-se sobre els models previs (jeràrquic i xarxa) durant la dècada dels vuitanta. Actualment és el model escollit per a la construcció de la majoria de sistemes de gestió de bases de dades comercials. El motiu del seu èxit resideix, per una banda, en la seva senzillesa, l'usuari percep la base de dades com un conjunt de "taules", i, per altra banda, en el caràcter declaratiu dels seus llenguatges de consulta i manipulació. La idea

fonamental és la utilització de "relacions"; pensant en cada relació com si fos una taula que està composta per registres (cada fila de la taula seria un registre o tupla), i columnes (també anomenades camps).

- **Base de dades relacional.** Conjunt finit de dades organitzades mitjançant taules, les quals es relacionen entre sí, i responen a unes condicions d'integritat determinades. En general, podem dir que una base de dades relacional, emmagatzema les dades en taules, formades per files i columnes, i que aquestes taules es poden unir amb la finalitat de recuperar dades relacionades que han estat emmagatzemades en diferents d'aquestes taules.

### Característiques

- Una base de dades relacional es compon de diverses taules o relacions.
  - No poden existir dues taules amb el mateix nom.
  - Cada taula és un conjunt de registres, files o tuples.
  - Cada registre representa un objecte del món real.
  - Cada un d'aquests registres consta de diverses columnes, camps o atributs.
  - No poden existir dues columnes amb el mateix nom en una mateixa taula.
  - Els valors emmagatzemats en una columna han de ser del mateix tipus de dada.
  - Totes les files d'una mateixa taula posseeixen el mateix nombre de columnes.
  - No es considera l'ordre en que s'emmagatzemen els registres a les taules.
  - No es considera l'ordre en que s'emmagatzemen les taules a la base de dades.
  - La informació pot recuperar-se o emmagatzemar-se per mitjà de sentències anomenades consultes.
- **Taula.** Conjunt de dades organitzades lògicament en columnes i files, en forma de matriu bidimensional.
  - **Camps (columnes).** Cada columna d'una taula en una base de dades, rep el nom de camp. Les taules obtenen el seu significat depenent dels camps que contenen, i per cada camp es descriuen les dades que poden emmagatzemar.

- **Registres (files).** Cada fila de la taula significa una instància de l'element que representi aquesta, i té tantes característiques com camps haguem definit en la creació d'aquesta taula. Els registres de les taules, es solen identificar amb entitats úniques, amb la qual cosa no solen haver-hi dos registres iguals.
- **Restriccions.** Una restricció és una condició que obliga el compliment de certes condicions en la base de dades. Algunes no les determinen els usuaris, sinó que són inherents al fet que la base de dades sigui relacional. Les restriccions proveeixen un mètode d'implementar regles a la base de dades. Les restriccions restringeixen les dades que es poden emmagatzemar en les taules. Habitualment es defineixen utilitzant expressions que donen com a resultat un valor booleà, indicant si les dades satisfan la restricció o no.  
Les restriccions no són part formal del model relacional, però s'hi inclouen perquè juguen el rol d'organitzar millor les dades.
- **Dominis.** Un domini descriu un conjunt de possibles valors per un cert atribut. Com que un domini restringeix els valors de l'atribut, pot ser considerat com una restricció. Són tipus de dominis: enters, cadenes de text, dates, etc.
- **Claus.** Una clau és un camp d'una taula que està indexada perquè qualsevol cerca sobre aquesta sigui més eficient. Cada taula pot contenir un o més camps els valors dels quals identifiquen de forma única cada registre de l'anomenada taula; és a dir, no poden existir dos o més registres diferents que tinguin el mateix valor en aquest camp. A aquest conjunt de camps se l'anomena **clau única**.  
Poden existir diverses claus úniques en una determinada taula, i a cada una d'aquestes se la sol anomenar candidata a clau primària.
- **Clau primària.** És una clau única elegida entre totes les candidates, per especificar les dades que seran relacionades amb les altres taules. La forma de fer això és per mitjà de claus forànies.  
Només pot existir una clau primària per taula i cap camp d'aquesta taula pot contenir valors NULL.

- **Clau forània.** És una referència a una clau en una altra taula. Les claus forànies no necessiten ser claus úniques.
- **Índex.** Els índexs s'utilitzen en les bases de dades perquè l'accés a les taules sigui més ràpid. Són llistes ordenades en les que és més fàcil trobar un element dins d'una taula. Els índexs es poden crear amb qualsevol combinació de camps en una taula. Les consultes que filtren registres per mitjà d'aquests camps, poden trobar els registres de forma no seqüencial utilitzant la **clau índex**.
- **Estructura.** La base de dades s'organitza en dues marcades seccions: l'esquema i les dades (o instància).
  - L'**esquema** és la definició de l'estructura de la base de dades i, principalment, emmagatzema les següents dades:
    - El nom de cada taula
    - El nom de cada camp
    - El tipus de dada de cada camp
    - La taula a la qual pertany cada camp
  - Les **dades** o **instància** és el contingut de la base de dades en un moment donat. És, en si, el contingut de tots els registres.
- **Relacions.** Un cop definides les taules en una base de dades, pot ser que algunes d'aquestes estiguin relacionades entre sí. Una relació consisteix en un lligam que pot establir-se entre dues taules que posseeixin alguns camps amb valors idèntics.

Existeixen certes regles que s'han de respectar a l'hora de crear bases de dades relacionals:

- **Les files són independents.** Cap fila d'una taula pot dependre de cap altra de la mateixa taula, per tant no té sentit cap ordenació de registres dins la mateixa.
- **Les files han de ser úniques.** Dues files d'una mateixa taula han de diferir almenys en el valor d'una columna.
- **Les columnes són independents.** Anàlogament al que passa amb les files, les columnes tampoc posseeixen cap ordre, això és, no té

cap significació des del punt de vista del model una eventual ordenació de les mateixes en la seva presentació.

- **Els valors de les columnes han de ser unitaris.** Dins d'una mateixa fila, cada columna, contindrà un valor, mai una fila o una llista d'ells.

La funcionalitat del model de bases de dades relacionals es basa en bona mesura en la duplicació de la informació. Així doncs, els dos (o més) camps implicats en una relació són la clau principal o primària, ja presentada anteriorment, i la **clau externa o forània**. La clau externa o forània és la clau de la taula relacionada que emmagatzema una còpia de la clau principal en la taula principal. Des del moment en que existeixen claus forànies, resulta evident que existeixen dades redundants entre taules diferents. No obstant, no està permès que les dades es repeteixin en una mateixa taula, com acabem de veure en les regles bàsiques sobre files i columnes.

Podem definir tres tipus de relacions entre taules en una base de dades relacional:

- **Un a un (1:1).** Un registre de la taula A no pot tenir més d'un registre coincident a la taula B, i per un registre de la taula B succeeix el mateix amb la taula A. Aquest tipus de relació és poc freqüent. En aquestes relacions, una taula normalment pren el lloc d'un camp d'una altra.
- **Una a molts (1:n).** Un registre de la taula A pot tenir més d'un registre coincident a la taula B; en canvi, un registre de la taula B té com a màxim un registre coincident amb la taula A. És el tipus més freqüent en les bases de dades relacionals. En aquest tipus de relacions, cada registre d'una taula pot tenir cap, un o diversos registres en una taula relacionada. Per aconseguir aquesta relació en el disseny de la nostra base de dades, hem de copiar la clau principal de la part única de la relació a la taula que emmagatzema la part de diversos de la relació. S'utilitzen per casos en els que existeixen valors repetits i són les relacions més comuns.
- **Molts a molts (n:m).** Un registre de la taula A pot tenir més d'un registre coincident amb la taula B, i un registre de la taula B també pot tenir més d'un registre coincident amb la taula A. Aquestes



relacions són una mica diferents, doncs requereixen d'una taula intermitja que lligui a les dues taules objecte d'estudi.

La definició correcta de les relacions entre les entitats que s'expressen a les taules són la base d'un disseny eficient d'una base de dades relacional.

- **Sistema de Gestió de Bases de Dades.** És una eina de software que permet la creació i manipulació de bases de dades. Un sistema de gestió de bases de dades és una aplicació propòsit general que permet crear bases de dades de qualsevol dimensió i complexitat i finalitats específiques diferents.

## 2.6.2 MySql

MySQL és un sistema de gestió de bases de dades relacional; una de les característiques més interessants de MySQL és que permet recórrer a bases de dades multiusuari a través de la web i en diferents llenguatges de programació que s'adaptin a diferents necessitats i requeriments. Per altra banda, MySQL és coneguda per desenvolupar alta velocitat en la cerca de dades i informació, a diferència de sistemes anteriors. MySQL és utilitzat en plataformes (Linux/Windows-Apache-MySQL-PHP), i la seva popularitat com a aplicació web està molt lligada a PHP, que sovint apareix combinat amb MySQL.

MySQL AB –des del gener de 2008 una subsidiària de Sun Microsystems– desenvolupa MySQL com software lliure en un esquema de llicenciament dual.

## 2.7 Eines de desenvolupament

### 2.7.1 Adobe Dreamweaver

És una aplicació destinada per a la construcció i edició de llocs i aplicacions web basats en estàndards. És el programa d'aquest tipus més utilitzat en el sector del disseny de pàgines web, per les seves funcionalitats i la seva integració amb altres eines, així com pel seu gran poder d'ampliació i personalització.

A més, Adobe Dreamweaver té les funcions típiques d'un editor de codi font per a la web:

- Un administrador de llocs, per agrupar els arxius segons al projecte al qual pertanyin.
- Un client FTP integrat, que permet pujar els arxius editats immediatament al lloc a Internet.
- Funció d'autocompletar i resaltar la sintaxis per instruccions en HTML i llenguatges de programació com PHP, JSP o ASP.

### 2.7.2 Adobe Fireworks

És una aplicació en forma d'estudi, però més semblant a un taller destinat pel maneigament híbrid de gràfics vectorials amb gràfics en mapa de bits i que ofereix un ambient eficient, tant per a la creació ràpida de prototips de llocs Web i interfícies d'usuari com per a la creació i la optimització d'imatges per webs. Està dissenyat per integrar-se amb altres productes de Adobe, com Dreamweaver.

## 2.7.3 Client ftp Filezilla

Filezilla és un programa que permet la transferència d'arxius al nostre servidor web utilitzant FTP (File Transfer Protocol). Aquest programa està dissenyat per suportar les màximes funcionalitats, cuidant sempre la velocitat i assegurant un entorn estable. Algunes de les principals característiques que presenta són la capacitat per continuar descàrregues interrompudes (sempre que el servidor ho suporti), possibilitat de treballar amb múltiples connexions o connexions a través de firewalls, i permet posar els arxius a pujar o descarregar a la cua.

## 2.7.4 PhpMyadmin

PhpMyAdmin és una eina escrita en PHP amb la intenció de manipular l'administració de MySQL a través d'una interfície web molt intuïtiva.

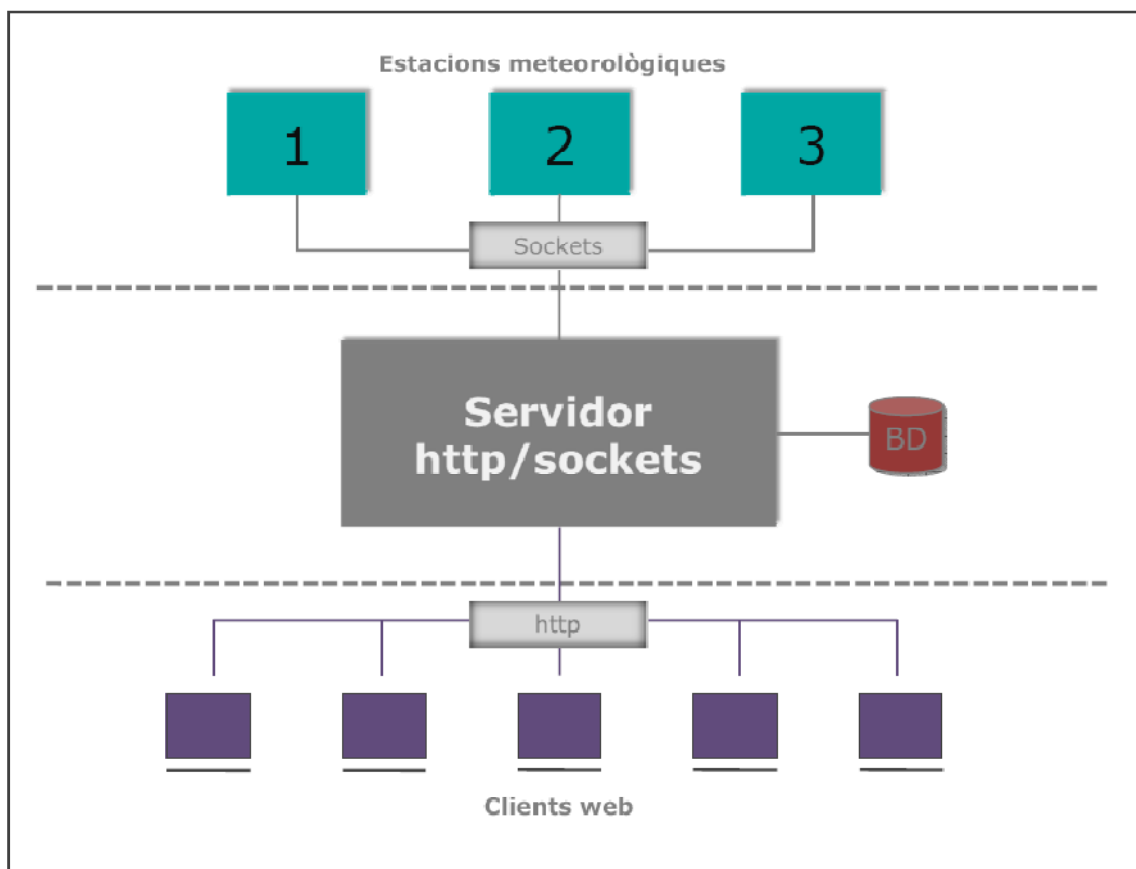
Aquesta aplicació ens permet crear i eliminar Bases de Dades, crear, eliminar i alterar taules, esborrar, editar i afegir camps, executar qualsevol sentència SQL, administrar claus en camps, navegar pels registres de les taules, editar-los i esborrar-los, administrar privilegis, exportar dades en diversos formats, i fer un backup de la base de dades.

Podem observar algunes d'aquestes accions amb tots els seus passos (crear bases de dades, crear taules, inserir registres, etc.), en la part d'annexos d'aquest projecte.

# 3 Desenvolupament projecte

## 3.1 Definició

En la figura següent, es mostra l'esquema intern del nostre projecte MeteoService. En la mateixa, es poden observar els tres components principals: les estacions meteorològiques, els clients web i el servidor, així com la base de dades utilitzada per emmagatzemar tots els valors de les variables meteorològiques (la temperatura, la humitat relativa i les precipitacions) i el tipus de comunicació que hi ha entre les diferents parts.



Tal i com ja hem anat comentant, es tracta d'un servei meteorològic que, accessible des d'Internet, proporciona lectures dels paràmetres de 3 estacions i, a més, queda dotat de dades històriques des de la seva posta en marxa. Totes aquestes dades les proporcionen les estacions meteorològiques, situades suposadament a Súria, Roses i Sort, simulant l'obtenció de cada una de les variables meteorològiques, i enviant-les al servidor via sockets per tal que aquest les guardi a la base de dades per a consultes d'usuaris a través de la pàgina web preparada amb aquesta finalitat.

A continuació es detallaran totes i cadascuna de les parts de les quals es compon aquest projecte, així com quines eines i mètodes s'han utilitzat per constituir-les i fer que funcioni:

### 3.1.1 Parts

#### Servidor HTTP Apache

El servidor HTTP Apache utilitzat en aquest projecte és part d'un allotjament web (en anglès *web hosting*) del tipus LAMP (Linux, Apache, MySQL i PHP), amb servidor FTP inclòs. Un allotjament web és, simplement, un servei que proveeix als usuaris d'Internet un sistema per poder emmagatzemar informació, imatges, o qualsevol contingut accessible via Web.

La tasca del servidor HTTP Apache és buscar el fitxer que peticona l'usuari de MeteoService a través d'un navegador web, interpretar-lo (es tracta de codi php que accedeix a base de dades), i retornar-lo al navegador en format HTML per tal que se li mostri per pantalla a l'usuari.

#### Servidor de Sockets

El servidor de sockets d'aquest projecte es manté a l'espera de peticions per part de les estacions meteorològiques. Aquest servidor té un socket obert sempre i prepara't per rebre les dades dels seus clients.

Des del servidor HTTP Apache s'ha programat un "cron job" que manté el servidor de sockets escoltant permanentment. Si hi ha qualsevol error amb el servidor, el

cron el torna a activar. Quan aquest rep una connexió d'alguna de les estacions, en separa la trama rebuda i emmagatzema les variables meteorològiques a la base de dades per tal que puguin ser consultables a partir del mateix moment.

## Estacions meteorològiques

Les estacions meteorològiques d'aquest projecte, que proporcionen les variables perquè els clients web les puguin consultar, estan simulades per 3 fitxers php (tants com estacions hi ha) que exerceixen de clients de sockets i fan un enviament de valors al servidor a cada hora.

Les estacions realitzen el paper de clients del servidor de sockets, que és qui emmagatzema les dades que les pròpies estacions li transmeten a cada comunicació. Per fer que les estacions exerceixin de clients, s'ha programat un "cron job" al servidor perquè cada hora executi els fitxers estacio1.php, estacio2.php, i estacio3.php. D'aquesta manera, cada hora en punt, que és quan s'ha planificat d'obtenir les dades, el client estacio1.php (i l'estacio2.php i estacio3.php) obté els valors de les variables meteorològiques d'aquella hora, mitjançant unes funcions de simulació que explicarem amb detall més endavant, i els envia al servidor perquè aquest els emmagatzemi a la base de dades de MeteoService.

## Clients Web

Tal i com ja hem anat comentant, un client web (navegador) és un programa mitjançant el qual l'usuari sol·licita a un servidor web l'enviament d'informació, informació que es transfereix mitjançant el protocol HTTP. Les dades rebudes són un conjunt de documents de text codificats en llenguatge HTML, que és el llenguatge que el navegador sap interpretar i mostrar a l'usuari per pantalla.

En aquest projecte el navegador que hem utilitzat és l'Internet Explorer de Microsoft, tot i que alternativament s'ha treballat també amb el Mozilla Firefox.

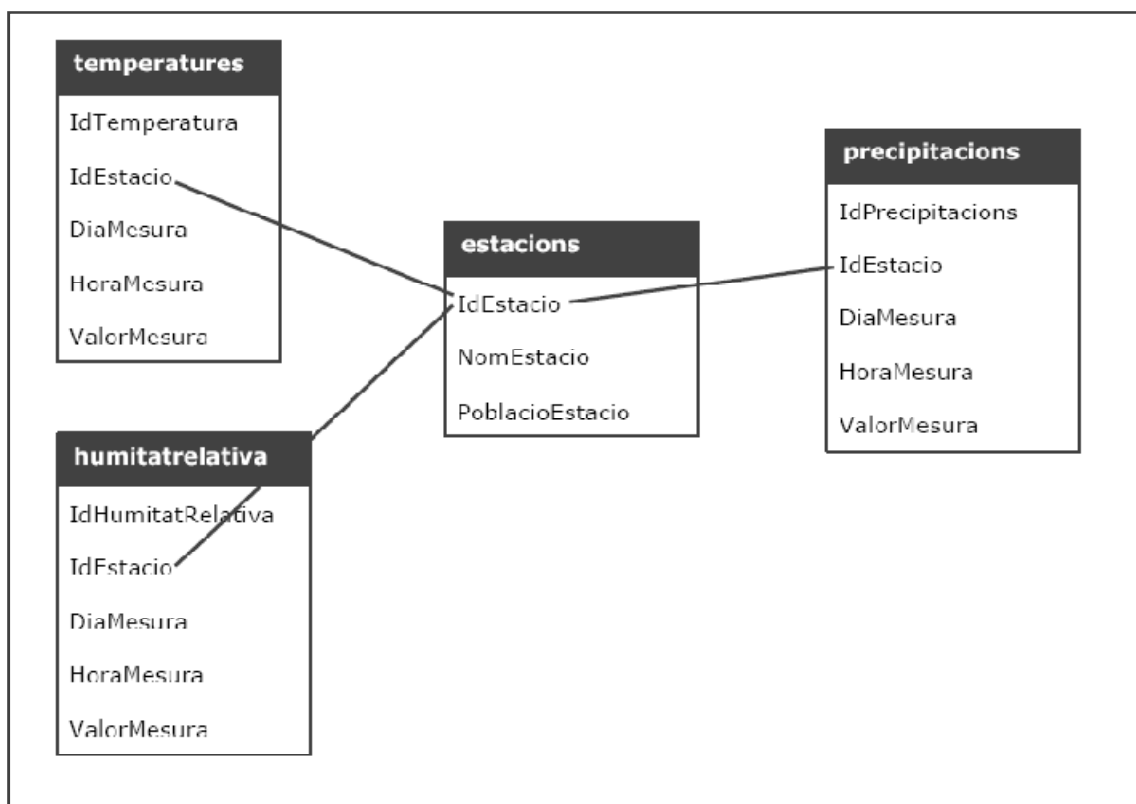
En el nostre projecte MeteoService, aquests navegadors mostraran per pantalla la web que permet escollir una estació concreta, així com consultar les dades meteorològiques del dia i hora que l'usuari desitgi.

## Base de dades del sistema

### Estructura bbdd Meteoservice

En aquest projecte s'ha implementat una bases de dades per tal d'organitzar tota la informació, que haurem de manipular i dividir en diferents taules. L'esquema de les relacions entre les taules de MeteoService, que veiem a la figura X, ens ajudarà a comprendre l'organització de la informació que plantejem.

En aquesta figura, podem apreciar clarament, que identifiquem cada entitat de la base de dades per un camp numèric anomenat Id (IdEstacio, IdTemperatura, IdHumitatRelativa i IdPrecipitacions), i que s'assigna automàticament en el moment d'afegir un nou registre a la taula indicada. Aquest Id és la clau primària de cadascuna de les taules, i a més l'utilitzem com a clau forània en les relacions 1 a diversos que hi ha entre algunes taules (estacions a temperatures, estacions a humitat relativa i estacions a precipitacions). Aquest sistema de pas d'informació mitjançant la relació entre claus de les taules evita que hi hagi dades repetides i que haguem d'introduir la informació més d'una vegada a la base de dades.



## Taules del sistema

A continuació farem una breu descripció de la informació emmagatzemada en cadascuna de les taules de la base de dades de MeteoService, que serveix com a estructura de dades pel projecte desenvolupat.

**Estacions.** Aquesta taula emmagatzema les dades de les Estacions de MeteoService. A cada estació li correspon un registre d'aquesta taula.

Camps	Descripció	Tipus dada	Clau/ Índex
IdEstacio	Número únic que identifica cada estació meteorològica	INT (11)	PRIMÀRIA
NomEstacio	Nom de l'estació meteorològica	VARCHAR(30)	-
PoblacioEstacio	Població de l'estació meteorològica	VARCHAR(30)	-

**Temperatures.** Aquesta taula emmagatzema les dades de les temperatures de cadascuna de les estacions meteorològiques de MeteoService.

Camps	Descripció	Tipus dada	Clau/ Índex
IdTemperatura	Número únic que identifica cada registre de temperatures	INT (11)	PRIMÀRIA Auto_increment
IdEstacio	Número únic que identifica cada estació meteorològica	INT (11)	ÍNDEX FORÀNIA
DiaMesura	Dia de la temperatura obtinguda	DATE	ÍNDEX
HoraMesura	Hora de la temperatura obtinguda	TIME	ÍNDEX
ValorMesura	Valor de la temperatura obtinguda	DECIMAL(10,1)	-



**Humitat Relativa.** Aquesta taula emmagatzema les dades de la humitat relativa de cadascuna de les estacions meteorològiques de MeteoService.

Camps	Descripció	Tipus dada	Clau/ Índex
IdHumitatRelativa	Número únic que identifica cada registre d'Humitat relativa	INT (11)	PRIMÀRIA Auto_increm ent
IdEstacio	Número únic que identifica cada estació meteorològica	INT (11)	ÍNDEX FORÀNIA
DiaMesura	Dia de l'obtenció de la variable meteorològica humitat relativa	DATE	ÍNDEX
HoraMesura	Hora de l'obtenció de la variable meteorològica humitat relativa	TIME	ÍNDEX
ValorMesura	Valor de la humitat relativa	FLOAT	-

**Precipitacions.** Aquesta taula emmagatzema les dades de les precipitacions de cadascuna de les estacions meteorològiques de MeteoService.

Camps	Descripció	Tipus dada	Clau/ Índex
IdPrecipitacions	Número únic que identifica cada registre de precipitacions	INT (11)	PRIMÀRIA Auto_increm ent
IdEstacio	Número únic que identifica cada estació meteorològica	INT (11)	ÍNDEX FORÀNIA
DiaMesura	Dia de les precipitacions obtingudes	DATE	ÍNDEX
HoraMesura	Hora de les precipitacions obtingudes	TIME	ÍNDEX
ValorMesura	Valor de la precipitacions obtingudes	DECIMAL(10,1)	-

Si analitzem la bases de dades de MeteoService a nivell de rendiment, molt probablement hauria estat més òptim guardar totes les variables en una taula. És a dir, fer que la base de dades de MeteoService constés de dues taules, una per les Estacions, i l'altra per les variables meteorològiques; per cada registre identificat pels índexs autonumèrics "IdVariable", IdEstacio, data i hora, hi hauria un valor de cada una de les variables meteorològiques: temperatura, humitat relativa i precipitacions, en els tres camps següents del mateix registre. A més, en els accessos a base de dades que realitza el servidor http quan consulta els valors amb els paràmetres peticionats pel client, o quan el servidor emmagatzema les dades un cop les estacions li envien els valors, reduiríem aquests accessos de 3 a 1, en aquest cas.

El motiu principal de no haver fet la distribució de la base de dades del projecte d'aquesta manera és la previsió de possibles ampliacions del mateix. Si en un futur es volen incorporar noves variables meteorològiques, que poden emmagatzemar-se amb tempos o paràmetres diferents (màxims i mínims, etc.) o que, senzillament, se'ls vol fer un tractament diferent, tenir una sola taula per a totes les variables ens hauria limitat. En canvi, una taula per cada variable ens ajuda tant a nivell de tractament com a nivell visual per entendre l'estructura, separada i amb possibilitat de forma d'explotació diferent per cada una de les variables.

Així doncs, en el projecte treballem inicialment amb tres variables meteorològiques que, tot i tenir tractaments semblants i emmagatzemar-se amb la mateixa freqüència, les mostrem en tres taules diferents.

També entenem que aquest projecte no està enfocat a l'optimització de bases de dades ni aplicacions informàtiques i que fer tres consultes i tres insercions no té una penalització tant costosa com per haver-ho de fer amb una, donat que accedim a cada una de les taules de variables meteorològiques per 3 índexs (IdEstacio, Data i Hora) i la mida de les mateixes tampoc és extremadament gran.

## Comunicació estacions-servidor

A l'hora de plantejar el projecte, necessitàvem un tipus de comunicació entre les estacions i el servidor que emmagatzemaria les dades, on és pogués manipular i parametritzar tot el funcionament i regles en la transmissió de les dades, així com que, preferentment, apliquéssim la filosofia client/servidor i, sobretot, que en les dades enviades i rebudes no hi hagués possibilitat d'error.

Tal i com ja hem comentat anteriorment, els sockets no són més un eina que habilita la comunicació i permet l'intercanvi de dades entre dos programes que poden o no estar situats en diferents ordinadors, utilitzant tres recursos (que a la vegada els defineixen): una adreça IP, un número de port i un protocol.

Quan utilitzem sockets, cal ser conscient que són un mecanisme de baix nivell, ja que hem d'interpretar els fluxos de dades que rebem. És a dir, que si necessitem que un servidor proveeixi molta funcionalitat, pot ser que se'ns compliqui utilitzant sockets. En canvi, si hi ha poca funcionalitat i molta transferència de dades, ens convenen més els sockets. Per altra banda, els sockets ens ofereixen més flexibilitat sobre la comunicació, podent establir a més baix nivell el tipus de protocol a utilitzar, i altres detalls de com es transmeten les dades.

Resumint, tenim els següents avantatges i inconvenients d'utilitzar sockets en la nostra comunicació:

### **Avantatges**

- Els sockets són ràpids, i introdueixen poca sobrecàrrega a les aplicacions.
- Són molt populars i difosos, pràcticament tots els llenguatges de programació i protocols els suporten.

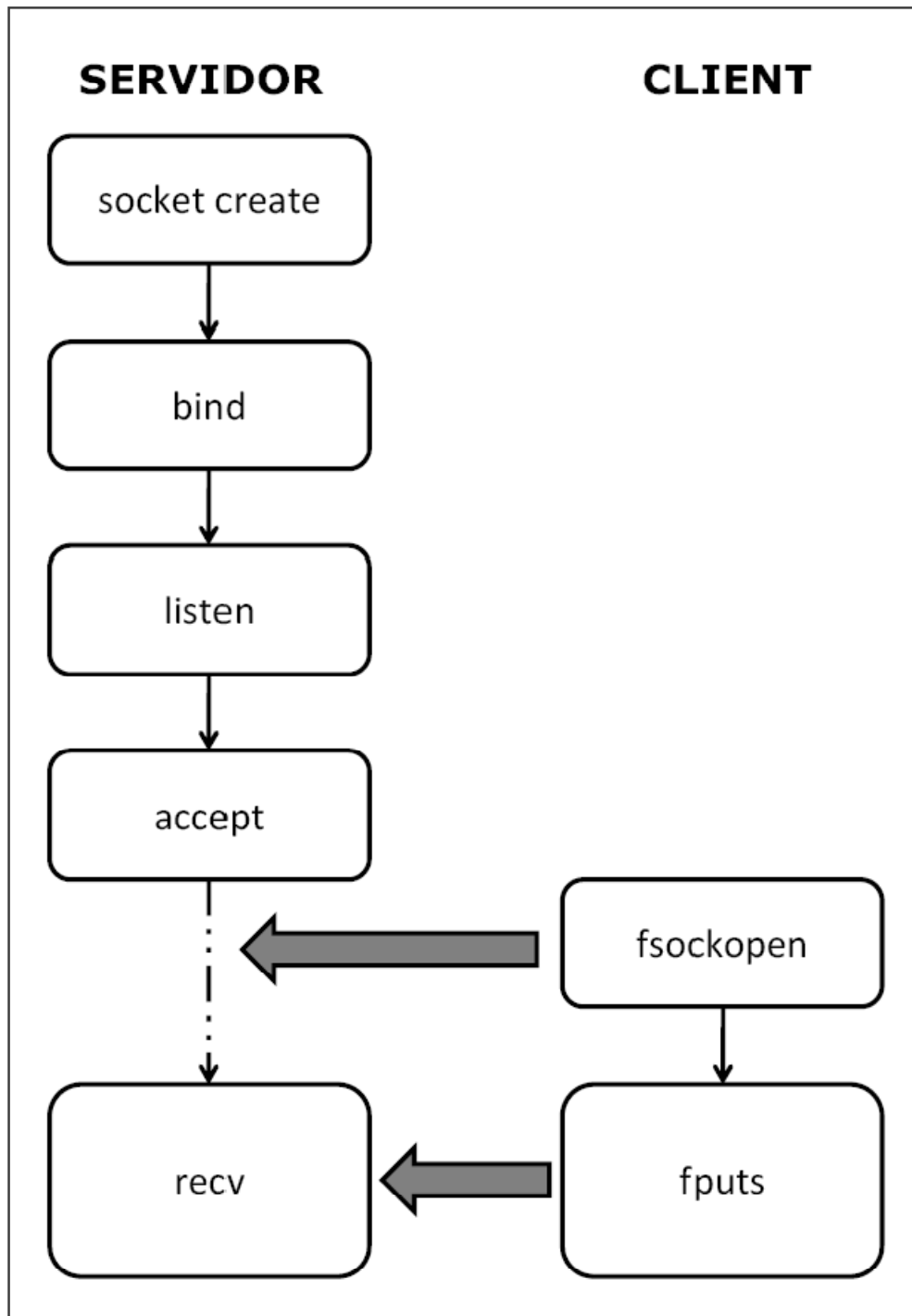
### **Inconvenients**

- Els sockets, al ser de baix nivell, no resulten gaire còmodes pel programador. Al no permetre el pas directe d'arguments, el programador ha d'encarregar-se d'obrir/tancar els fluxes de I/O, col·locar-hi els arguments que vol passar, extreure'n els resultats, etc.
- El codi és difícil de reutilitzar.

Així doncs, donat que ja m'interessava controlar i definir les trames que transmeten els valors de les variables meteorològiques així com poder entendre i saber interpretar tots i cadascun dels punts de la comunicació, vaig decidir que els sockets serien un bon mitjà de transmissió de dades que compleix totes les

expectatives que tenia previstes per a la comunicació entre les estacions i el servidor.

A continuació, en la següent figura, es mostra un esquema de la connexió i instruccions per a la comunicació via sockets de MeteoService:



El tipus de socket que utilitzem és el `SOCK_STREAM`, sockets orientats a connexió, que permeten una comunicació bidireccional, garanteixen l'entrega segura dels paquets i la informació ordenada al destí, utilitzant el protocol TCP/IP.

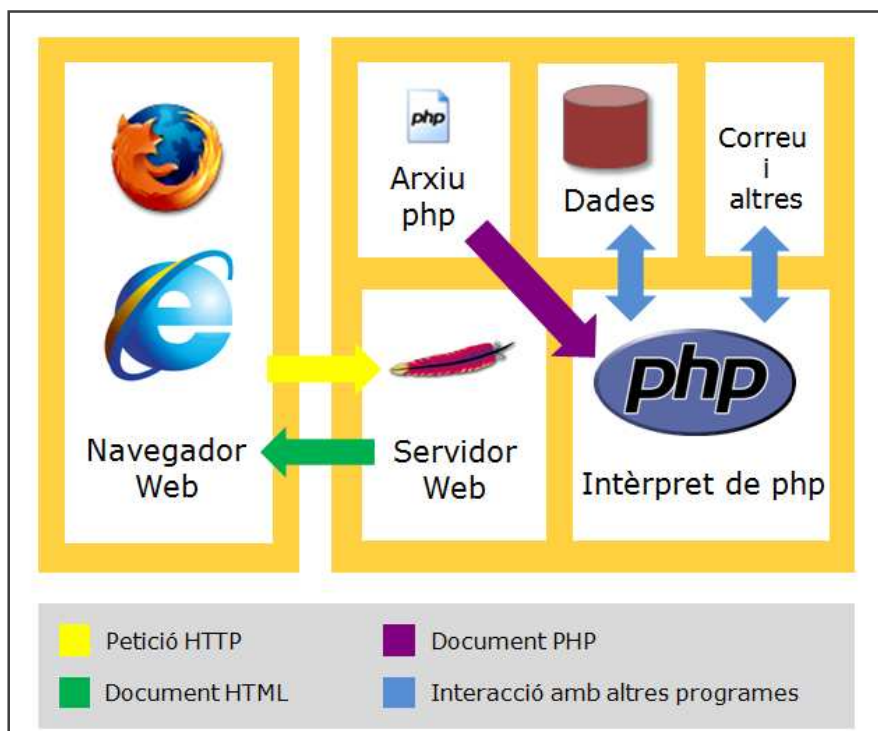
Les instruccions mostrades en la figura, que executen el servidor i el client de sockets per comunicar-se i transmetre's dades, així com els paràmetres que requereixen cadascuna d'elles, es detallaran als annexos d'aquest projecte.

## Comunicació clients web-servidor

La comunicació entre els clients web i el servidor es realitza mitjançant el protocol HTTP. Tal i com ja hem anat comentant anteriorment, la comunicació comença quan l'usuari tecleja la URL de la pàgina inicial, en aquest cas la de MeteoService, i el navegador la tradueix en una adreça IP per tal de fer una connexió TCP amb el servidor web a través del port 80, el predeterminat pels serveis web.

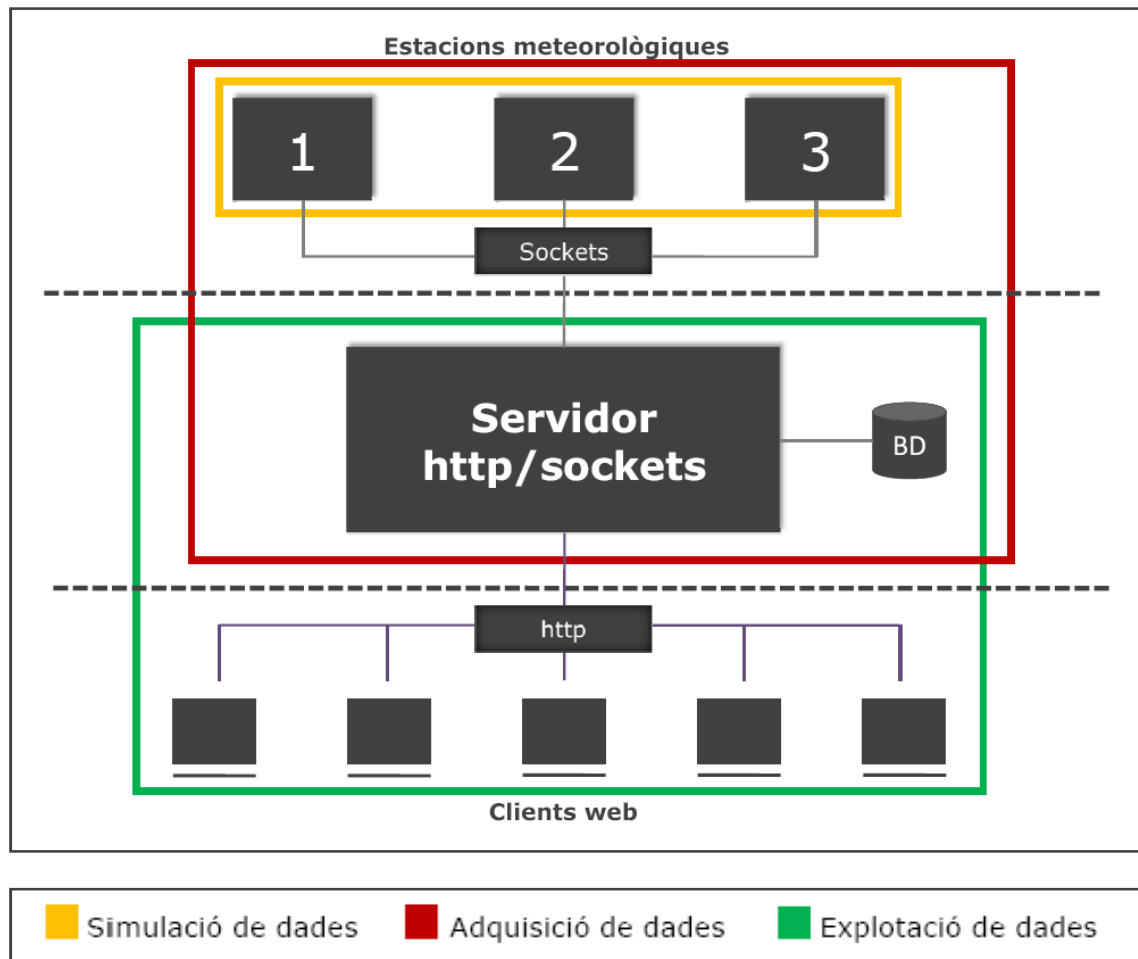
Quan la connexió s'ha establert, el navegador envia una petició HTTP al servidor web sol·licitant la pàgina indicada teclejada per l'usuari. A continuació, el servidor executa l'interpret de PHP. Aquest processa l'script sol·licitat que generarà el contingut de manera dinàmica (en el nostre cas obtenint informació d'una base de dades). El resultat és enviat per l'interpret al servidor, qui, a la vegada, envia al client en format HTML, prèviament havent tramès una resposta estàndard HTTP, indicant si la petició ha tingut èxit o no.

La figura següent mostra el funcionament descrit:



## 3.2 Operativa

Si prenem l'esquema general del projecte presentat anteriorment, el podem dividir en tres parts ben diferenciades pel què fa al tractament de les dades:



La primera és l'origen dels valors de les variables meteorològiques, i està formada per les tres estacions del projecte i les funcions simuladores que generen aquests valors.

La segona és l'adquisició de les dades meteorològiques i la inserció de les mateixes a la base de dades de MeteoService.

La tercera és l'explotació de les dades, que es tradueix en les pantalles que veu l'usuari a través d'Internet i des d'on es poden consultar els valors d'aquestes variables meteorològiques.

## 3.2.1 Simulador de dades

Tal i com ja hem anat comentant amb anterioritat, els valors de les variables meteorològiques són simulats i generats per rutines que donen un valor aleatori a les mateixes. Aquest projecte està pensat perquè des de la seva posta en marxa quedi dotat de dades històriques consultables per l'usuari web. És a dir, cada hora les 3 estacions simulen un valor per cada una de les variables meteorològiques (temperatura, humitat relativa i precipitacions) i els envien al servidor, que s'encarregarà d'insertar-los a la base de dades de MeteoService, indexats pel dia i l'hora de la mesura, i deixar-los accessibles per posteriors consultes a través d'Internet.

A continuació explicarem el funcionament de les estacions meteorològiques i les rutines que criden per obtenir els valors de les variables:

### estacio1.php

---

L'estació obre un socket al servidor mitjançant el port 9000, port que no té cap servei predefinit i que hem decidit que seria l'utilitzat en la comunicació via sockets. Si la connexió no genera cap error, es pot continuar amb el procés, obtenint la data i hora del sistema i cridant a les tres rutines generadores de valors meteorològics passant-les-hi com a paràmetres aquesta data i hora.

Seguidament, un cop obtingudes les dades de les funcions cridades, es prepara una trama per enviar al servidor mitjançant el socket obert. El format de la trama que hem creat per a la comunicació amb el servidor i pas de valors és el següent:

#### TRAMA:

```
ESTACIO * DATA * HORA * VALOR_TEMPERATURA *  
VALOR_HUM_REL * VALOR_PRECIPITACIONS
```

on

**ESTACIO** és l'IdEstacio de l'estació connectada

**DATA** és la data actual del sistema

**HORA** és l'hora actual del sistema

**VALOR\_TEMPERATURA** és el valor que ha generat la funció simuladora de temperatures

**VALOR\_HUM\_REL** és el valor que ha generat la funció simuladora d'humitat relativa

**VALOR\_PRECIPITACIONS** és el valor que ha generat la funció simuladora de precipitacions

\* és el separador estàndard escollit per diferenciar els diversos valors de què consten les trames que es transmeten en les comunicacions entre les estacions i el servidor

Quan l'estació ha generat la trama, l'envia al servidor mitjançant el socket obert i, si tot ha succeït de forma correcta, es tanca el socket i es finalitza el procés de l'estació, fins que a la següent hora comenci altra vegada.

A continuació es detallarà el funcionament de les rutines generadores de valors:

## Funcions generadores de valors

---

### • Funció generador\_temperatures

Per realitzar la funció de generació de valors de temperatures, inicialment s'han recollit temperatures històriques d'una estació meteorològica real, i s'han introduït en matrius multidimensionals. Per cada primer dia de cada mes de l'any 2008 s'ha obtingut la temperatura mitja de cada hora i s'ha emmagatzemat aquest valor  $\pm 1^{\circ}\text{C}$ ; és a dir, no s'ha guardat la temperatura mitja, sinó la temperatura mitja  $-1^{\circ}\text{C}$ , com a temperatura mínima, i la temperatura mitja  $+1^{\circ}\text{C}$ , com a temperatura màxima, valors que s'utilitzaran en la funció per generar-ne d'aleatoris per cada dia i hora.

Així doncs, l'estructura de la matriu bidimensional de temperatures és la següent:

```
$temp ['mes']['hora'] = 'temperatura_mínima|temperatura_màxima'
```

on

`mes` és l'identificador numèric de cada mes (01 gener, 02 febrer, ..., 12 desembre)

`hora` és l'hora del dia en format HH

`temperatura_mínima` és la temperatura mitja extreta de l'hora, del dia 1 del mes indicat menys  $1^{\circ}\text{C}$



| és el separador que utilitzem per diferenciar el valor de la temperatura mínima i la màxima

`temperatura_màxima` és la temperatura mitja extreta de l'hora, del dia 1 del mes indicat més 1 °C

Inicialment, la funció generadora de temperatures extreu el número de mes de la data passada per paràmetre en format AAAA-MM-DD. A continuació, es guarda en una variable el valor de la matriu seleccionant-la pel mes obtingut i l'hora del paràmetre i se'n separa el valor mínim i màxim de temperatura, que seran els límits que s'utilitzaran per generar-ne un d'aleatori.

Aquest valor és el quin retorna la funció generadora de temperatures per la data i hora passades per paràmetre en cridar-la.

Per cada una de les estacions del nostre projecte, tot i que la climatologia dels llocs on simuladament estan situades, hem utilitzat les mateixes dades per generar els valors de les temperatures accessibles a posteriori per usuaris.

### • **Funció generador\_humitat\_relativa**

La funció de generació de valors d'humitats relatives funciona exactament de la mateixa forma que la de temperatures.

Per poder tenir dades a partir de les quals generar els valors d'humitats relatives simulades, s'han recollit dades històriques d'humitats relatives d'una estació meteorològica real, i s'han introduït en matrius multidimensionals. Per cada primer dia de cada mes de l'any 2008 s'ha obtingut la humitat relativa mitja de cada hora i s'ha emmagatzemat aquest valor +-2 punts percentuals; és a dir, no s'ha guardat la humitat relativa mitja, sinó la humitat relativa mitja -2%, com a humitat relativa mínima, i la humitat relativa mitja +2%, com a humitat relativa màxima, valors que s'utilitzaran en la funció per generar-ne d'aleatoris per cada dia i hora.

L'estructura de la matriu bidimensional és la següent:

```
$temp ['mes']['hora'] = 'hum_rel_mínima|hum_rel_màxima'
```

on

`mes` és l'identificador numèric de cada mes (01 gener, 02 febrer, ..., 12 desembre)

`hora` és l'hora del dia en format HH

`hum_rel_mínima` és la humitat relativa extreta de l'hora, del dia 1 del mes indicat menys 2 punts percentuals

| és el separador que utilitzem per diferenciar el valor de la humitat relativa mínima i la màxima

`hum_rel_màxima` és la humitat relativa extreta de l'hora, del dia 1 del mes indicat més 2 punts percentuals

Inicialment, la funció generadora d'humitats relatives extreu el número de mes de la data passada per paràmetre en format AAAA-MM-DD. A continuació, es guarda en una variable el valor de la matriu seleccionant-la pel mes obtingut i l'hora del paràmetre i se'n separa el valor mínim i màxim d'humitat relativa, que seran els límits que s'utilitzaran per generar-ne un d'aleatori.

Aquest valor és el que retorna la funció generadora d'humitats relatives per la data i hora passades per paràmetre en cridar-la.

Per cada una de les estacions del nostre projecte, tot i que la climatologia dels llocs on simuladament estan situades, hem utilitzat les mateixes dades per generar els valors de les humitats relatives accessibles a posteriori per usuaris.

## • Funció generador\_precipitacions

A la funció generador de valors de precipitacions se li passen els mateixos paràmetres, data i hora, que a les altres dues rutines, però el tractament de les dades per obtenir els valors de les precipitacions per cada dia i hora des de la posta en marxa del projecte, no és ben bé el mateix.

S'emmagatzema en una matriu que conté tots els mesos de l'any, la quantitat mínima i màxima de litres que hi poden haver en cas de precipitacions, extreïes d'estadístiques històriques.

L'estructura de la matriu en aquest cas és la següent:

```
$rain ['mes'] = 'precip_mínim|precip_màxim'
```

on

`mes` és l'identificador numèric de cada mes (01 gener, 02 febrer, ..., 12 desembre)

`precip_mínim` és la quantitat mínima de mil·límetres

| és el separador que utilitzem per diferenciar el valor mínim i màxim de quantitat de precipitacions

`precip_màxim` és la quantitat màxima de mil·límetres

Inicialment, la funció generadora de valors de precipitacions extreu el número del mes de la data passada per paràmetre en format AAAA-MM-DD. A continuació, es guarda en una variable el valor de la matriu seleccionant-la pel mes obtingut, i en se'n separa el valor de precipitacions mínim i màxim.

A continuació, s'inicia el mètode pròpiament per generar la quantitat de precipitacions de l'hora i dia concrets, i primerament s'obté un valor aleatori entre el mínim i el màxim. Si aquest valor és zero, la quantitat de precipitacions que la funció retornarà seran 0 mil·límetres. Si aquest valor és superior a 0, es decideix que plourà si la probabilitat és més del 50% (aquesta probabilitat s'obté de dividir el valor màxim de precipitacions i l'aleatori obtingut al principi). Si plou, la quantitat de precipitacions, en mil·límetres, que retornarà la funció, serà un valor aleatori entre el mínim i el màxim de la matriu de precipitacions

Per cada una de les estacions del nostre projecte, tot i que la climatologia dels llocs on simuladament estan situades, hem utilitzat les mateixes dades per generar els valors de les precipitacions accessibles a posteriori per usuaris.

Tal i com ja hem introduït en algun capítol anterior, la comunicació client-servidor que hi ha entre les estacions i el servidor de sockets, no es cenyeix a la teoria de la comunicació client-servidor ja que el client, les estacions meteorològiques en el nostre cas, exerceixen aquest paper d'una manera forçada des del propi servidor. D'aquesta manera fem que els clients facin la connexió al servidor cada hora en punt, trametent les dades de les tres variables meteorològiques perquè el mateix servidor les insereixi a la base de dades.

Cal remarcar, també, que l'exactitud i precisió dels valors de les dades meteorològiques a consultar no era un objectiu del projecte, sinó que la principal fita era la comunicació interna entre les diferents parts del mateix. Per tant, ens interessava tenir dades a consultar, mínimament coherents amb la climatologia del nostre país, segons època de l'any, així com hora del dia, però no era transcendent que fossin vertaderes perquè això no influència amb la finalitat d'aquest projecte. Així mateix, el projecte està pensat per possibles ampliacions, la primera de les quals podria ser la connexió al servidor d'estacions reals en comptes de simuladors.

### 3.2.2 Adquisició de dades

La part d'adquisició de dades d'aquest projecte està formada per les estacions i el servidor de sockets i, tal i com ja hem anat comentant, aquest últim és qui emmagatzema aquestes dades per a consultes posteriors d'usuaris a través d'Internet.

El servidor de sockets s'ha programat de tal manera que quan crea un socket, no en crea un de sol, sinó que el què fa és generar una matriu (array) de sockets. D'aquesta forma, quan, a cada hora en punt, les estacions meteorològiques fan una petició al servidor per enviar-li les dades, ja està preparat per gestionar totes aquestes trames i rebre de forma fiable i ordenada la informació de cada una de les estacions i introduir-la a la base de dades correctament.

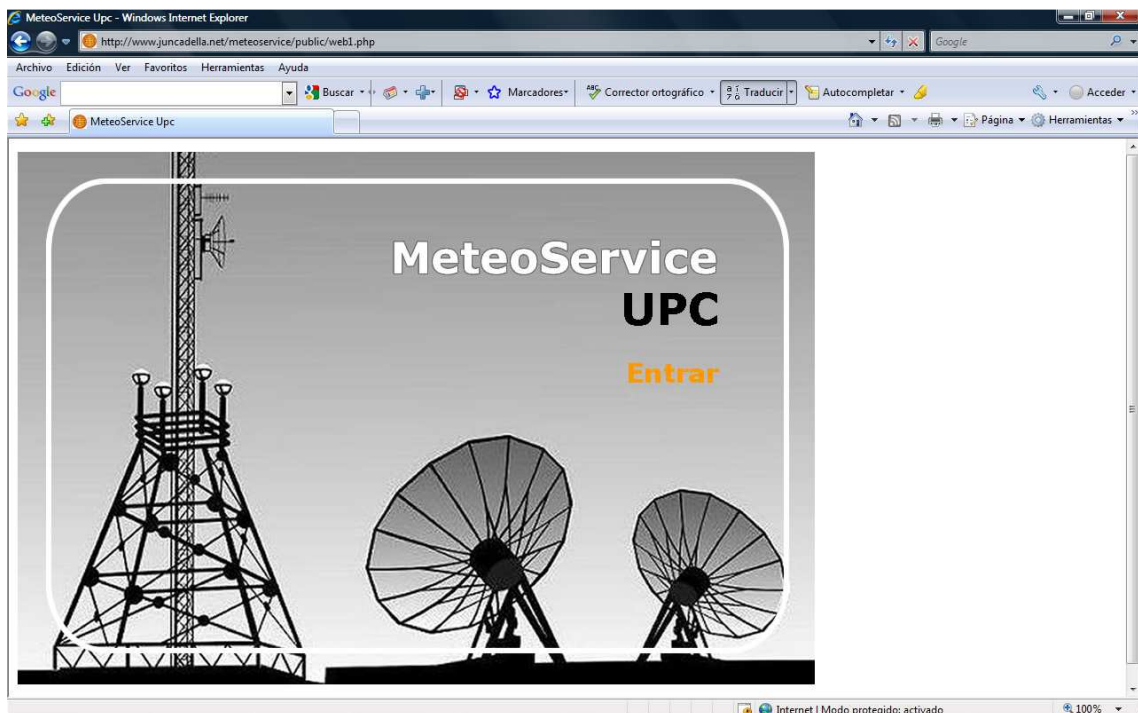
A nivell d'aquesta inserció de valors a les taules de MeteoService, el servidor, en rebre la trama d'una estació, la separa pel caràcter establert com a estàndard en aquest projecte per diferenciar les dades de les trames, mitjançant una funció pròpia de php. És llavors, un cop disgregada tota aquesta informació, que executa cadascuna de les instruccions (INSERT's) necessàries perquè el valor de la temperatura, humitat relativa i precipitacions, quedin correctament afegits a la base de dades, i relacionats amb l'estació de la qual en són dades.

### 3.2.3 Explotació de dades

L'explotació dels valors de les variables meteorològiques generats per les estacions i emmagatzemats a la base de dades es realitza a través d'Internet. L'usuari accedeix a la pàgina de MeteoService i des d'allà pot escollir de quina estació vol consultar dades meteorològiques, seleccionant un dia i una hora en concret.

A continuació mostrarem cada una de les pantalles realitzades per tal finalitat i explicarem els programes i mètodes utilitzats per realitzar-les, així com el funcionament intern i tractaments que tenen:

#### Web1:



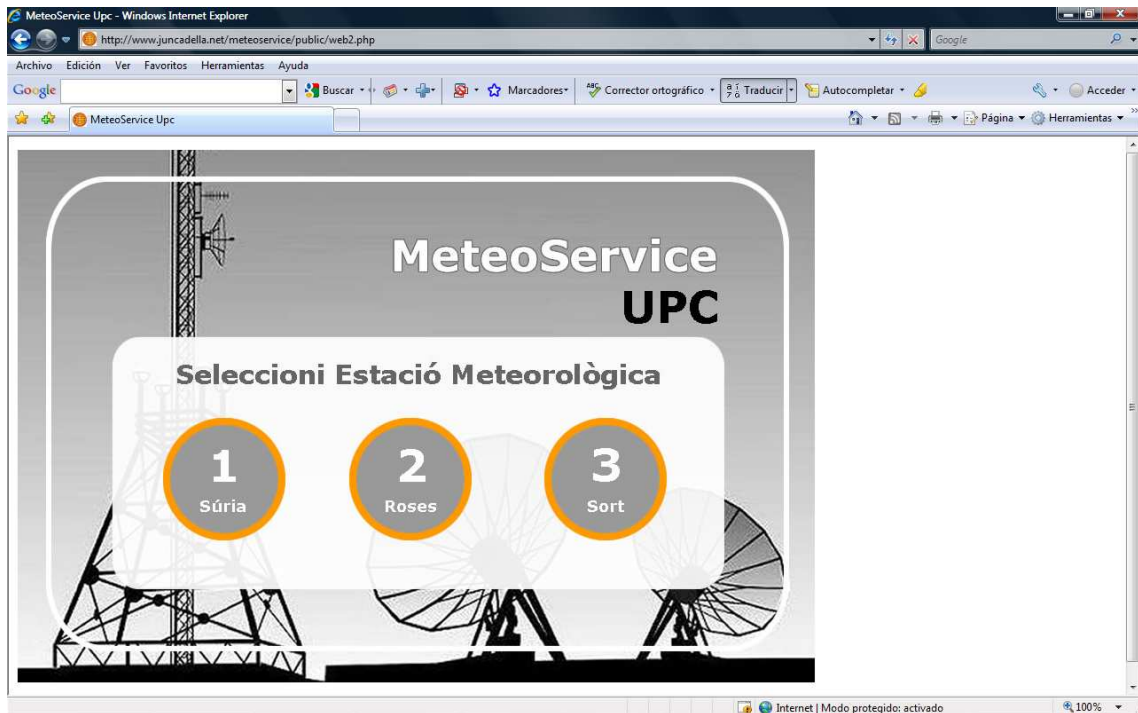
Aquesta és la pàgina que l'usuari visualitza quan introdueix la URL de MeteoService: [www.juncadella.net/meteoservice/public/web1.php](http://www.juncadella.net/meteoservice/public/web1.php).

En aquesta pantalla, que es tracta de la pàgina de presentació de MeteoService, l'usuari prem la zona activa **Entrar** i a continuació se li presenta la següent pàgina, web2, en pantalla.

Les zones actives són hipervincles, que poden tenir diferents formes i grandàries, a altres URL's. Per definir aquesta zona activa **Entrar**, s'ha seleccionat una àrea rectangular que englobi el títol Entrar i, a continuació, s'ha definit que quan es

premi al damunt de la zona, s'enllaci directament amb la URL de la web2 de MeteoService. D'aquesta manera, amb les zones actives, s'aconsegueixen enllaços més intuïtius i de més fàcil accés per l'usuari, al mateix temps que ajuden a millorar l'estètica general de la pàgina.

## Web2:

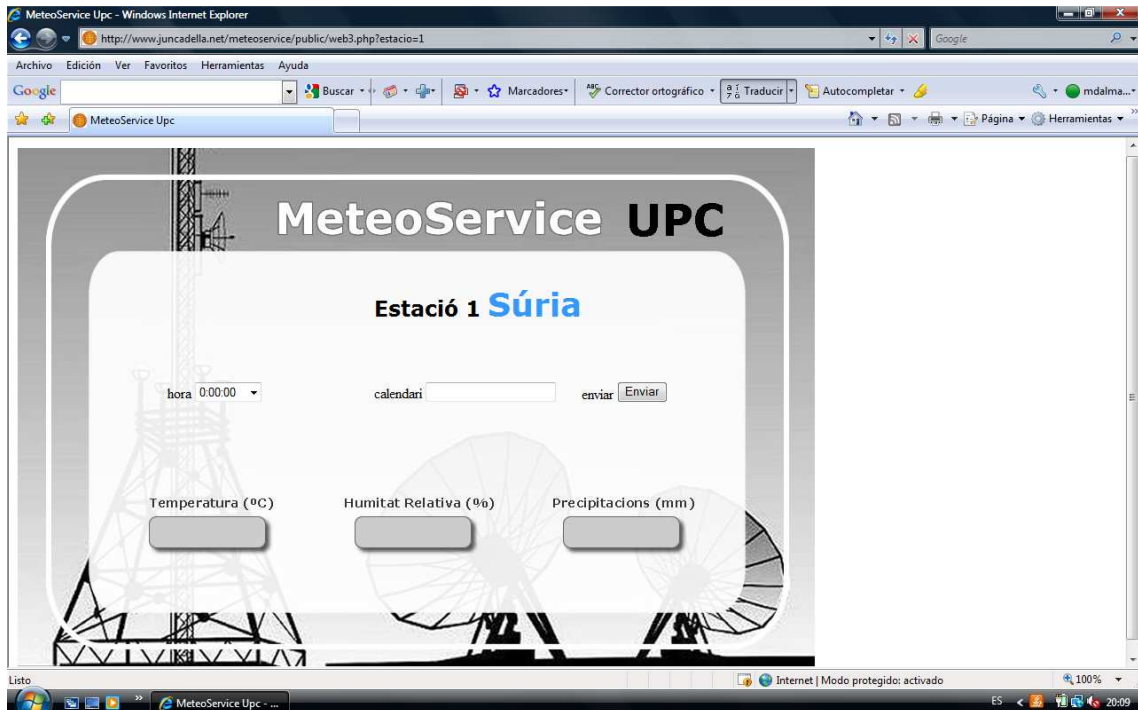


En aquesta segona pàgina, l'usuari pot escollir entre una de les tres estacions meteorològiques disponibles. En prémer damunt de la zona activa rodona de l'estació desitjada, anirem cap a la següent pàgina de consulta de dades.

En aquesta pàgina s'inicia una sessió. Recordem que una sessió és bàsicament la seqüència de pàgines que visita un usuari quan entra al nostre lloc web; des que entra al lloc fins que l'abandona. En el cas que ens ocupa, ens interessa saber la sessió en la que ens trobem per diferenciar l'estació escollida per l'usuari; és important per futures consultes a bases de dades tenir identificada en tot moment l'estació en la qual ens trobem.

Quan es prem damunt d'una de les tres zones actives, es guarda una variable de sessió, el valor de la qual serà el de l'estació seleccionada, que es transmetrà a la següent pàgina a través de la URL. Tal i com hem comentat, aquesta variable la podrem utilitzar durant tota la sessió per saber en quina estació ens trobem i de quina estació se'ns poden sol·licitar dades.

## Web3:



Aquesta és la pàgina a través de la qual l'usuari pot consultar els valors de les variables meteorològiques pel dia i l'hora prèviament seleccionats.

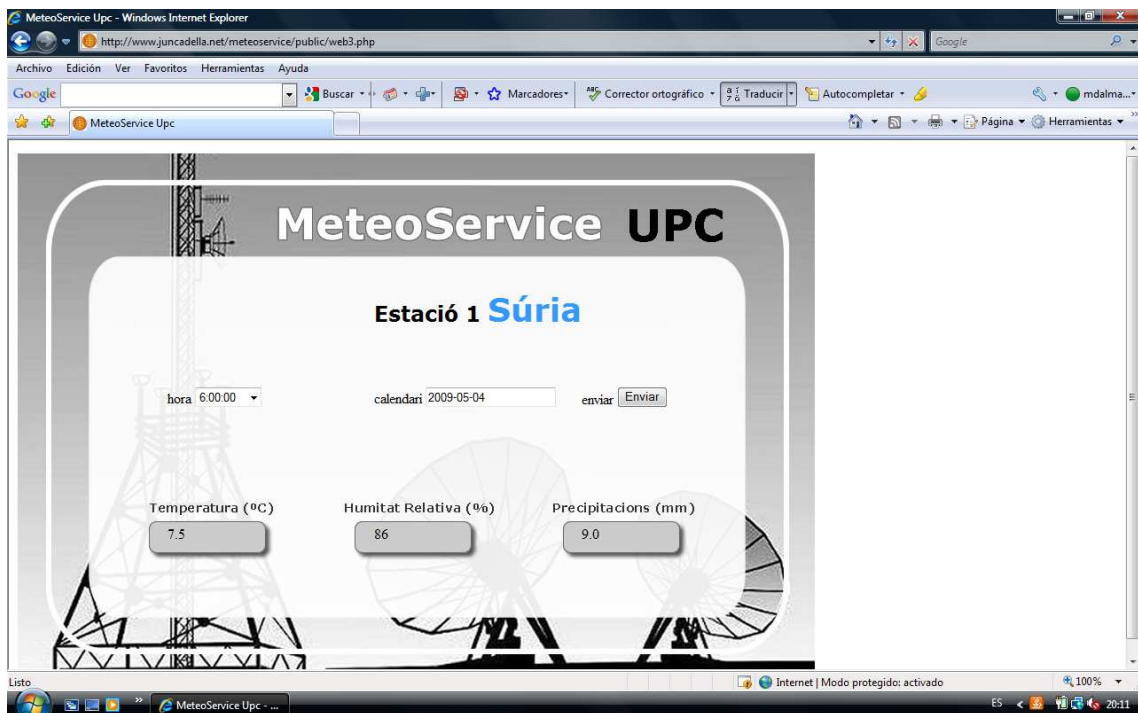
Tal i com hem comentat abans, en la pantalla anterior, quan l'usuari realitza l'acció de prémer la zona activa de l'estació escollida, s'inicia una sessió i s'envia l'identificador de l'estació a la pàgina web3, a través de la URL i de forma totalment transparent per l'usuari. (Observem com s'envia el número que identifica l'estació en la URL: [www.juncadella.net/meteoservice/public/web3.php?estacio=1](http://www.juncadella.net/meteoservice/public/web3.php?estacio=1)).

Abans que es carregui a la pantalla la web3, es recupera de la URL aquest identificador d'estació, mitjançant instruccions estàndards de php (`$_GET['estacio']`), i realitza una consulta a la base de dades de MeteoService, a la taula Estacions, per obtenir el nom de la població i mostrar-lo per pantalla en la presentació inicial de la pàgina.

Aquest identificador es guarda en una variable de sessió (`$_SESSION['estacio']`) i durant tot el temps que l'usuari estigui navegant i consultant dades en aquesta pàgina, és a dir, durant tota la sessió, tindrem a la nostra disposició aquesta variable que ens indica l'estació meteorològica actual i amb quina fer consultes contra la base de dades.

Un cop seleccionat el dia i l'hora de consulta, en prémer enviar, es recuperen aquests paràmetres, així com l'estació en la qual ens trobem i s'accedeix a les taules de les variables meteorològiques, Temperatures, Humitat Relativa i Precipitacions, per obtenir-ne els valors corresponents, en graus centígrads ( $^{\circ}\text{C}$ ), en percentatge (%) i en mm, respectivament.

A continuació, i per mostrar la consulta a l'usuari, l'aplicació imprimeix aquests valors per pantalla, tal i com es pot observar en la figura següent:



Per crear les imatges de fons de les pantalles de MeteoService s'ha utilitzat el programa Macromedia FireWorks, ideal per qualsevol tipus de tractament amb imatges, capes d'imatges i dissenys de pàgines web en general, així com per a la optimització per al rendiment posterior a la web.

Per programar tot el nostre projecte, tant a nivell de web, com a nivell de tractament de dades via sockets, com a nivell de funcions i rutines simuladores de valors, s'ha utilitzat el Macromedia DreamWeaver. Tal i com ja hem anat explicant, aquest programa és molt útil per a la construcció i edició d'aplicacions web amb contingut dinàmic, permetent la connexió a Bases de Dades com MySQL utilitzant tecnologia d'script, com PHP, i tot això d'una manera molt gràfica i intuïtiva per al programador.



# 4 Pressupost

---

## 4.1 Consideracions prèvies

Cal tenir en compte que aquest pressupost fa referència a un projecte de disseny d'una aplicació informàtica concreta. Llavors, és fàcil entendre que en la materialització d'aquest tipus de treballs no és necessària pràcticament la utilització de cap mena de material; únicament s'emptra l'habilitat i destresa del programador.

Es comptabilitzarà aquest pressupost tenint en compte les hores destinades a la realització de l'aplicació, separant-les de les hores necessàries per realitzar la memòria del projecte, tal i com ja comentarem més endavant. Cal remarcar que, en realitat, aquest pressupost és totalment teòric donat que es realitza desinteressadament ja que es tracta d'un treball de fi de carrera.

Pel què fa al Hardware i Software necessari pel desenvolupament, no el comptabilitzarem com un cost perquè ja disposem des d'un inici d'un PC preparat i de tots els programes (FireWorks, DreamWeaver, etc.) necessaris per a construir l'aplicació. L'únic cost econòmic al què s'haurà de fer front és al lloguer del servidor d'Internet, que ja especificarem més endavant.

## 4.2 Costos

**Determinació del cost/hora per categoria.** Cal tenir en compte els següents paràmetres:

- L'anàlisi funcional de l'aplicació l'ha realitzat el propi programador, conjuntament amb el professor tutor del projecte.
- No ha estat necessària la intervenció d'un analista de sistema: els programes utilitzats per realitzar aquesta aplicació els ha instal·lat el propi programador ja que es tracta d'aplicacions de nivell d'usuari.

- Per tant, doncs, el cost es determinarà exclusivament amb les hores emprades pel programador, tenint en compte que es tracta d'un programador júnior.

Consultant en el mercat sobre el preu actual de l'hora d'un programador júnior, obtenim un preu aproximat de:

PROGRAMADOR JÚNIOR ..... **30 euros per hora**

**Planificació i estudi dels temps de les diferents fases del projecte** (figura ja mostrada en la introducció del projecte):

<b>Planificació Temporal</b>																		
Fases/Activitats	Mesos															Hores		
	1	2	3	4	5	6												
<b>Fase 1: Anàlisi</b>																		
<b>Activitat 1.1.</b> ANÀLISI DELS REQUERIMENTS DE L'USUARI																	30	
<b>Activitat 1.2.</b> DOCUMENTACIÓ I EINES																	40	
<b>Activitat 1.3.</b> GENERACIÓ DE LA DOCUMENTACIÓ																	20	
<b>Fase 2. Disseny</b>																		
<b>Activitat 2.1.</b> DISSENY DE LA BASE DE DADES METEOSERVICE																	15	
<b>Activitat 2.2.</b> DISSENY DE LES FUNCIONS GENERADORES DE VALORS																	15	
<b>Activitat 2.3.</b> DISSENY DE LA INTERFÍCIE WEB																	30	
<b>Activitat 2.4.</b> DISSENY DEL MÒDUL DE SOCKETS																	35	
<b>Activitat 2.5.</b> GENERACIÓ DE LA DOCUMENTACIÓ																	25	
<b>Fase 3. Implementació i proves</b>																		
<b>Activitat 3.1.</b> IMPLEMENTACIÓ DE MÒDULS DE BASES DE DADES																	10	

[illegible]

A partir d'aquesta taula on es comptabilitzen totes les hores emprades pel programador, distingirem, per realitzar el pressupost final, entre les hores d'anàlisi, de disseny i d'implementació i proves, i les hores de generació de la documentació. Volem diferenciar-les perquè es tracta d'un projecte de final de carrera i el nivell de detall escrit en la memòria és més alt (s'hi inclouen també tots els conceptes bàsics), que el de la documentació que es podria generar si fos un projecte de veritat, que probablement constaria d'un anàlisi orgànic amb tot el detall tècnic de l'aplicació i el seu funcionament, però partint de la base que l'analista o programador ja té coneixement de la matèria.

En aquest pressupost final també tenim en compte el lloguer del servidor d'Internet HTTP Apache que emprem en l'aplicació, com a únic cost apart dels honoraris del programador.

Pressupost	Temps (hores)	Costos (euros)	Total (euros)
Programador	300	30	9000
Generació documentació	70	30	21
Lloguer del servidor d'Internet	-	75 €/any	75
<b>TOTAL</b>	<b>370</b>		<b>11175 €</b>

# 5 Ampliacions

---

La primera ampliació que ens podem plantejar en aquest projecte, i la més important i lògica, és la de portar-lo a la realitat fent que les dades que ens proporcionen les estacions siguin vertaderes i extretes d'estacions meteorològiques reals. D'aquesta manera, les dades que es podrien consultar a través de la mateixa pàgina web, serien reals i podríem conèixer, des de qualsevol punt del món amb connexió a Internet, la temperatura, humitat relativa i les precipitacions de Súria, Roses i Sort, i les  $n$  estacions més que es volguessin connectar.

Es podria aprofitar tot el software programat per adquirir les dades enviades per les estacions i emmagatzemar-les a la base de dades per a les consultes dels usuaris. Així mateix, les pàgines de consulta dels valors meteorològics podrien ser les mateixes i el seu funcionament exactament igual. Partiríem de la base, doncs, que totes les estacions meteorològiques estan connectades físicament a un PC i hi envien els valors de les variables periòdicament, assumint que les deixa accessibles des de la nostra aplicació, ja sigui en un fitxer o en una base de dades pròpia; si es donés aquest últim cas, es podria utilitzar un ODBC (*Open Data Base Connectivity*, Connectivitat Oberta de Bases de Dades), que és un estàndard d'accés a Bases de Dades que fa possible l'accés a qualsevol dada des de qualsevol aplicació, sense importar com estiguin emmagatzemades aquestes dades (MySQL, SQL Server, Access, etc.). En el PC de les estacions hi hauria instal·lat un servidor http Apache, amb php. I en el servidor, de la mateixa manera, es programaria un cron job perquè s'executés cada hora l'aplicació que enviaria, via sockets, les dades al servidor "central" del nostre projecte. Aquest és el quin continuaria tenint el control i gestionant la base de dades de MeteoService, que és la quina conté totes les dades de les estacions que el client web consultaria des d'Internet, ja obtenint valors reals de cada un dels llocs on estan situades les estacions.

Una altra de les ampliacions factibles de realitzar, i molt senzilla i adaptable al projecte actual al mateix temps, és la incorporació de noves variables meteorològiques per tenir més informació de la climatologia de l'estació. Algunes d'aquestes variables podrien ser, per exemple, la pressió atmosfèrica, mesurada en mil·libars (mb) o mil·límetres de mercuri (mm Hg), o la velocitat del vent

(expressat en km/h o m/sg) i direcció del vent (si indica Nord, significa que el vent va cap el Sud).

D'aquesta manera, caldria incorporar al nostre projecte tantes taules a la base de dades com noves variables meteorològiques decidim afegir, tenint en compte fer un estudi previ de amb quina freqüència ens interessa adquirir les dades, o quines característiques de la variable meteorològica ens interessa guardar per consultar posteriorment.

I, ja per finalitzar aquesta part de possibles ampliacions, es podrien afegir més funcionalitats a la pàgina web de MeteoService. Algunes d'aquestes incorporacions, podrien ser, per exemple, un apartat d'estadístiques que generessin gràfics de l'evolució de les variables meteorològiques, una nova pàgina on es poguessin comparar les dades de les diferents estacions, i fer gràfics creuats entre diferents variables, una part de gestió d'estacions només accessible amb usuari i password, i altres noves idees i ampliacions que es poden anar incorporant progressivament i sempre que es desitgi.

# 6 Conclusions

---

La finalitat d'aquest projecte és el disseny i desenvolupament d'un grup d'elements que, treballant conjuntament i utilitzant diverses formes de comunicació, generen, transmeten i emmagatzemen dades, per tal que siguin consultables quan l'usuari final així ho sol·liciti.

Tots aquests components i les interaccions que es generen entre ells, es tradueixen en el servei meteorològic que hem realitzat en el nostre projecte. Accessible des d'Internet, proporciona lectures simulades dels paràmetres meteorològics de 3 "estacions", suposadament instal·lades en tres poblacions diferents del nostre país. Els simuladors de les estacions es connecten via sockets al servidor http, que va generant la base de dades a mesura que va rebent els valors de les variables meteorològiques. El client web (usuari), pot consultar, des de qualsevol lloc amb connexió a Internet i en qualsevol moment, aquestes dades meteorològiques a través de la pàgina web preparada i creada per tal efecte.

Tal i com ja hem anat remarcant en el treball, l'enfocament del projecte no ha estat obtenir dades simulades molt exactes i semblants a la realitat de la situació climatològica de cada estació, sinó la comunicació que s'ha generat entre les diferents parts del projecte: els sockets per una banda, i la comunicació http per l'altra. D'aquesta manera s'han après mètodes de transmissió de dades i nous llenguatges de programació que es poden utilitzar per infinitat de processos i aplicacions diverses.

Per realitzar aquest treball hem seguit les fases de desenvolupament d'un projecte informàtic, determinant i tenint en compte sempre els requeriments de l'usuari, cercant i recollint documentació per obtenir diverses possibilitats i mètodes de desenvolupament de l'aplicació, en una primera fase. Dissenyar l'estructura i el funcionament de cada un dels components de l'aplicació, així com la realització de testos per verificar el funcionament individual dels mateixos, en una segona fase. I, finalment, aconseguir que tots els elements dissenyats arribin a funcionar en conjunt, fent els canvis, proves i ajustos necessaris per aconseguir-ho, en la tercera i última fase. Tots aquesta passos, tenint en compte que, per cada un d'ells, s'ha

generat la documentació que s'ha anat creient oportuna per finalment obtenir una memòria de projecte detallada i completa.

Tal i com entenem que és habitual que succeeixi en el procés de desenvolupament d'un projecte d'aquestes característiques, han anat sorgint alguns contratemps, que afortunadament s'han pogut superar.

El fet de realitzar una aplicació amb un llenguatge de programació i forma de comunicació del qual no se'n tenien coneixements previs, obliga forçosament a un aprenentatge inicial i constant durant tot el desenvolupament. Tot i que també es pot considerar una avantatge, obliga a dedicar molt més temps del què hi hauria dedicat un programador experimentat. Cal dir, però, que com que els llenguatges emprats són molt estesos i utilitzats arreu en l'actualitat, la cerca d'informació fiable i fàcil de comprendre no ha suposat cap problema ni contratemps. Així doncs, pel què fa al llenguatge php i a l'html, l'aprenentatge ha sigut relativament fluid, el mateix que ha succeït amb l'anàlisi i creació de la base de dades. Pel què fa a la comunicació via sockets i a la gestió del servidor, són potser les dues parts més desconegudes i difícils d'entendre per mi, i la cerca de manuals i informació ha estat més costosa que en les parts anteriors, però finalment s'ha aconseguit l'aprenentatge necessari i suficient per a la realització del projecte.

Un altre problema relacionat amb el temps disponible és que s'ha hagut de compaginar la realització del projecte amb una feina de jornada completa, sense flexibilitat horària. Aquest, però, és un obstacle que s'ha superat fàcilment amb molta força de voluntat i dedicació a les hores lliures.

Apart dels problemes esmentats, era previsible trobar-nos amb contratemps de naturalesa diversa durant el desenvolupament del projecte, tant relacionats amb el llenguatge escollit, el sistema de comunicació, la gestió del servidor, com en l'anàlisi i confecció de la base de dades. Tots ells, però, s'han pogut solucionar de sense més transcendència.

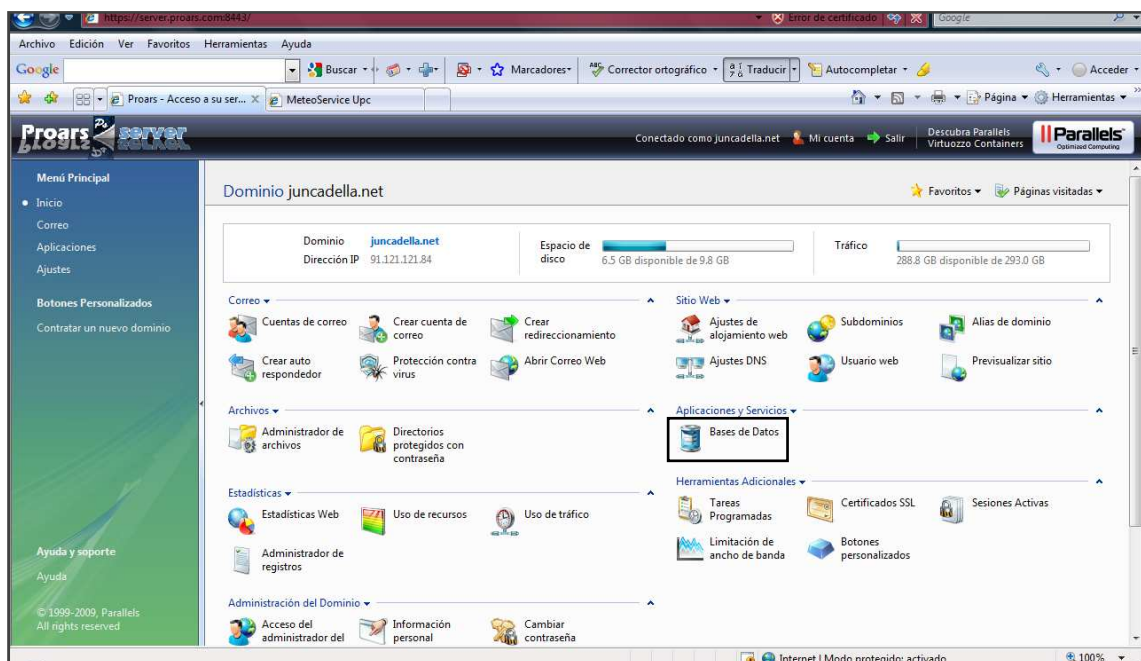
Així doncs, podem dir que tots els objectius inicials del projecte han estat satisfets, i totes les expectatives generades inicialment han estat cobertes. Crec, a més, que els coneixements adquirits poden servir-me en la meva carrera professional, i obrir-me més portes en l'entorn laboral, ja que aquest projecte ha augmentat el meu grau de coneixement i cultura sobre les aplicacions i comunicacions a través d'Internet i, el més important, m'ha ajudat molt perquè m'ha donat la base per saber com buscar solucions eficients i ràpides als problemes que poden anar sorgint pel camí.

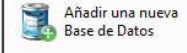
# 7 Annexes

## 7.1 Bases de Dades amb phpMyAdmin


En el següent apartat mostrarem els passos que cal seguir per crear una Base de Dades i totes les taules que la formen, així com la realització de consultes i insercions a les mateixes.

Des del servidor web, premem **Bases de Datos** de l'apartat "Aplicaciones y Servicios".



Premem  per crear una nova base de dades, introduint el nom al formulari que se'ns mostra per pantalla:



A continuació, accedim a phpMyAdmin seleccionant  i se'ns mostra la següent pantalla:

En primer damunt del nom de la base de dades, ens dóna l'opció de Crear una nova taula:

A continuació, introduïm els diversos Camps dels quals constarà la taula, i definim les característiques que tindran (el tipus (INT, FLOAT, VARCHAR, etc), si es tracta d'un índex, etc):

127.0.0.1 / localhost / meteoservice / estacions | phpMyAdmin 3.1.1 - Windows Internet Explorer

http://127.0.0.1/phpmyadmin/index.php?db=meteoservice&token=44088aed59087fb4e2bc17d7bf506d9

Base de datos: meteoservice (1)

Servidor: localhost | Base de datos: meteoservice | Tabla: estacions

Campo	IdEstacio		
Tipo	INT	INT	INT
Longitud/Valores <sup>1</sup>			
Predeterminado <sup>2</sup>	Ninguna	Ninguna	Ninguna
Cotejamiento			
Atributos			
Nulo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Índice	---	---	---
A_I	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comentarios			

Comentarios de la tabla:

definición de la PARTICIÓN:

Motor de almacenamiento: MyISAM

Cotejamiento:

Grabar O Añadir 1 campo(s) Continuar

Un cop definit, premem **Grabar**. Ens dóna l'opció, al costat del botó de gravar, d'afegir els n camps més que necessitem, si pel motiu que sigui no ho hem fet abans:

127.0.0.1 / localhost / meteoservice / estacions | phpMyAdmin 3.1.1 - Windows Internet Explorer

http://127.0.0.1/phpmyadmin/index.php?db=meteoservice&token=44088aed59087fb4e2bc17d7bf506d9

Base de datos: meteoservice (1)

Servidor: localhost | Base de datos: meteoservice | Tabla: estacions

Campo	IdEstacio	NomEstacio	PoblacioEstacio
Tipo	INT	VARCHAR	VARCHAR
Longitud/Valores <sup>1</sup>			
Predeterminado <sup>2</sup>	Ninguna	Ninguna	Ninguna
Cotejamiento			
Atributos			
Nulo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Índice	UNIQUE	---	---
A_I	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comentarios			

Comentarios de la tabla:

definición de la PARTICIÓN:

Motor de almacenamiento: MyISAM

Cotejamiento:

Grabar O Añadir 1 campo(s) Continuar

Internet | Modo protegido: activado

Podem observar la taula amb tots els camps i característiques de cadascun d'ells:

Server: localhost:3306 Base de datos: ps1203\_meteoservice Tabla: estaciones

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar Eliminar

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> IdEstacio	int(11)			No			
<input type="checkbox"/> NomEstacio	varchar(30)	latin1_swedish_ci		No			
<input type="checkbox"/> PoblacioEstacio	varchar(30)	latin1_swedish_ci		No			

Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Vista de impresión Vista de relaciones Planteamiento de la estructura de tabla

Añadir 1 campo(s) Al final de la tabla Al comienzo de la tabla Después de IdEstacio Continuar

Índices:				Espacio utilizado	
Nombre de la clave	Tipo	Cardinalidad	Acción	Tipo	Uso
PRIMARY	PRIMARY	3		Datos	68 Bytes
Crear un índice en 1 columna(s) Continuar				Índice	2,048 Bytes
				Total	2,116 Bytes

Estadísticas de la fila	
Enunciado	Valor
Formato	dinámico/a
Cotejamiento	latin1_swedish_ci
Filas	3
Longitud de la fila	22
Tamaño de la fila	705 Bytes
Creación	08-06-2009 a las 20:17:16

Per començar a inserir registres (en aquesta taula hi hem introduït els registres manualment, donat que inicialment, el projecte, només consta de 3 estacions), premem la pestanya **Insertar**, i omplim els valors dels camps.

127.0.0.1 / localhost / meteoservice / estaciones | phpMyAdmin 3.1.1 - Windows Internet Explorer

http://127.0.0.1/phpmyadmin/index.php?db=meteoservice&token=44088aed59087fb4e2bc1d7bf506d9

Base de datos: meteoservice (1)

meteoservice (1)

Insertar

Campo	Tipo	Función	Nulo	Valor
IdEstacio	int(11)		1	
NomEstacio	varchar(30)			Sant Salvador
PoblacioEstacio	varchar(30)			Súria

Continuar

☐ Ignorar

Campo	Tipo	Función	Nulo	Valor
IdEstacio	int(11)		2	
NomEstacio	varchar(30)			Platja d'Aro
PoblacioEstacio	varchar(30)			Platja d'Aro

Continuar

Insertar como una nueva fila y luego Volver

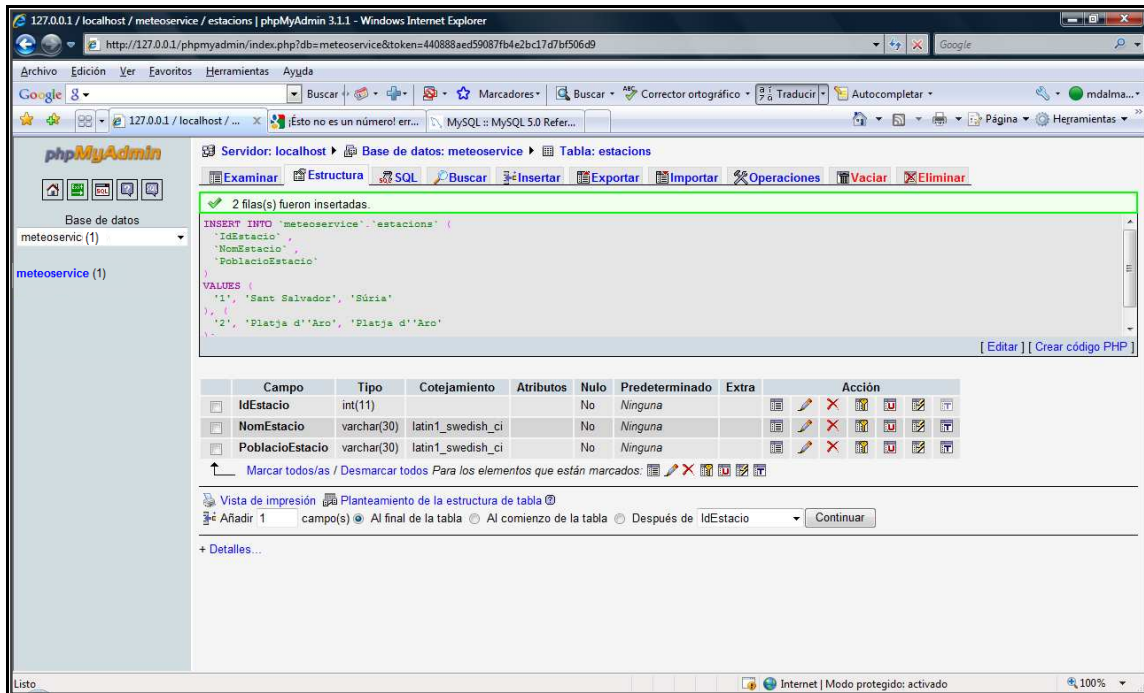
1 Continuar Reiniciar

Reinicie la inserción con 2 filas

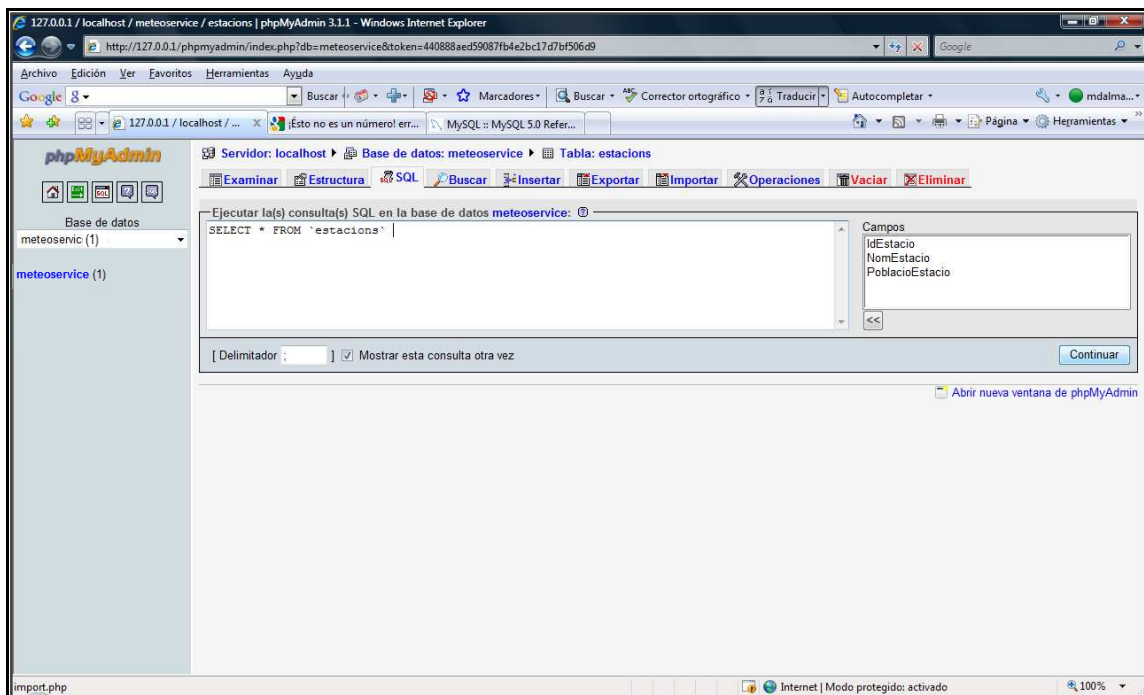
1 Use la tecla TAB para saltar de un valor a otro, o CTRL+flechas para moverse a cualquier parte

Internet | Modo protegido: activado

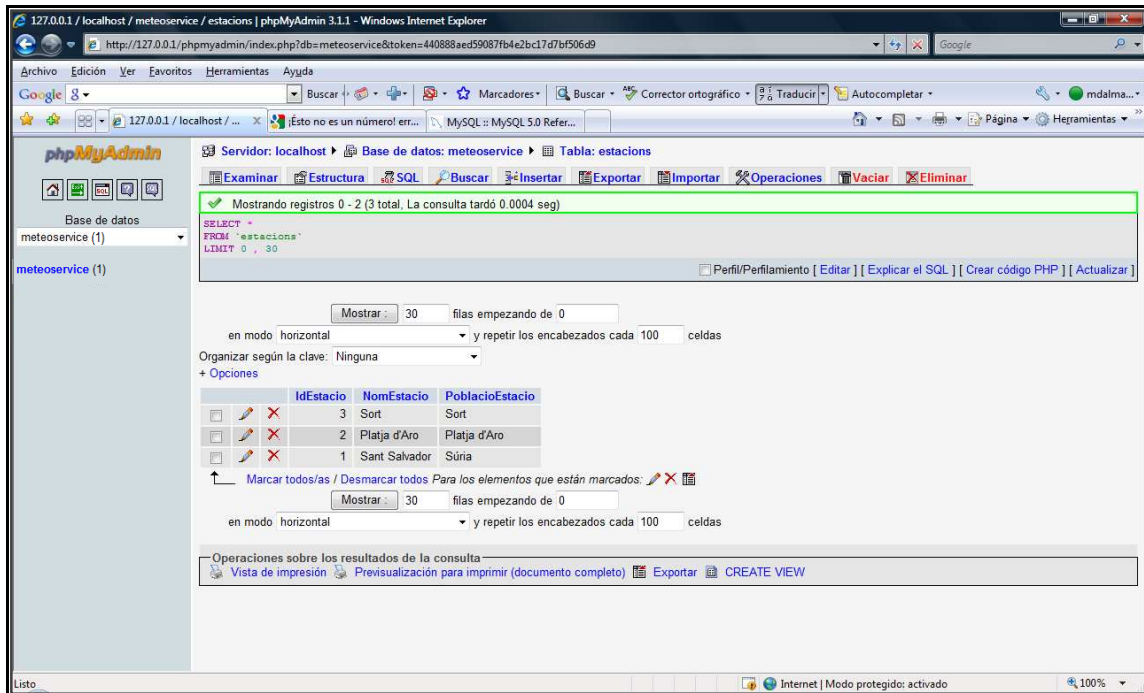
En prémer Continuar, podem observar com se'ns mostra la sentència SQL que s'ha executat en omplir els camps.



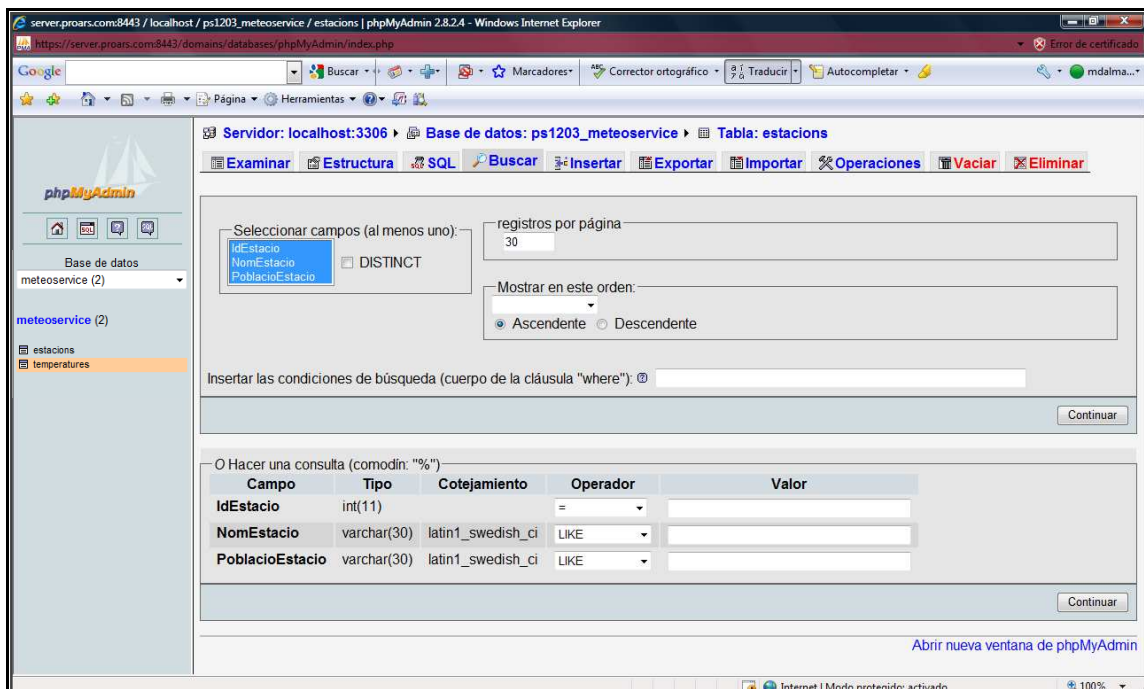
Podem fer que se'ns mostrin tots els registres de la taula, o fer una SELECT amb les condicions que vulguem determinar, introduint la sentència SQL directament des de la pestanya **SQL** (en aquest cas fem una consulta de tots els registres que hi ha en aquesta taula, sense cap tipus de filtre):



Se'ns mostren tots els registres (i la sentència SQL a la part superior):

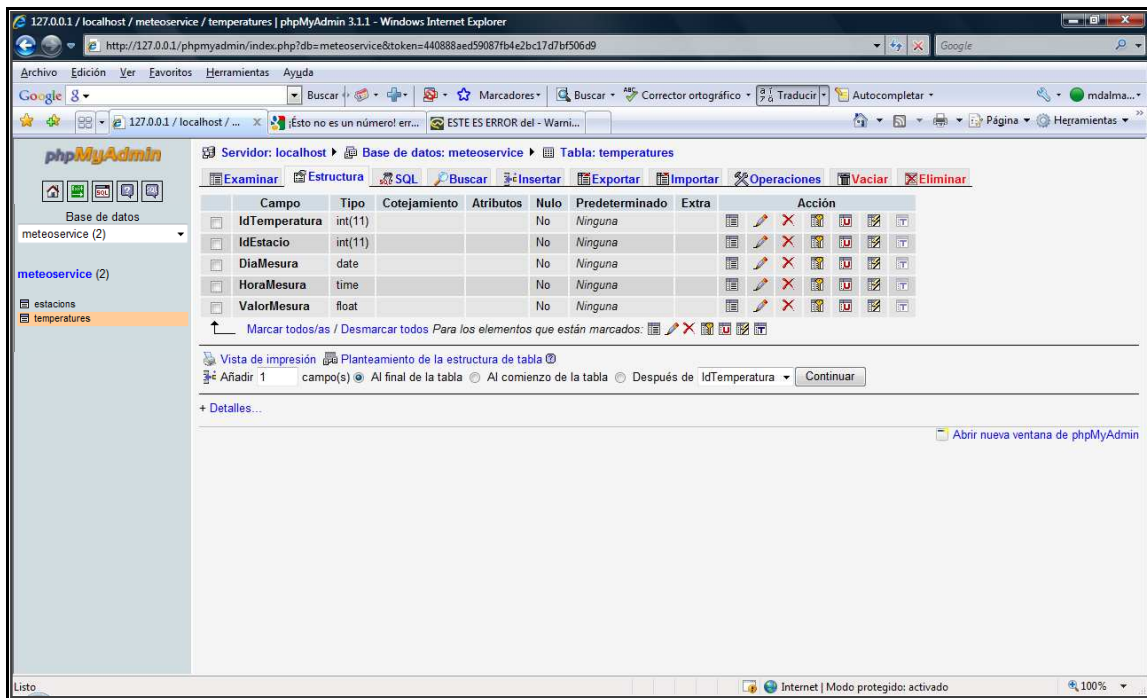


També, a la pestanya **Buscar**, tenim l'opció de fer una cerca a la taula, sense haver d'introduir la sentència SQL amb codi:

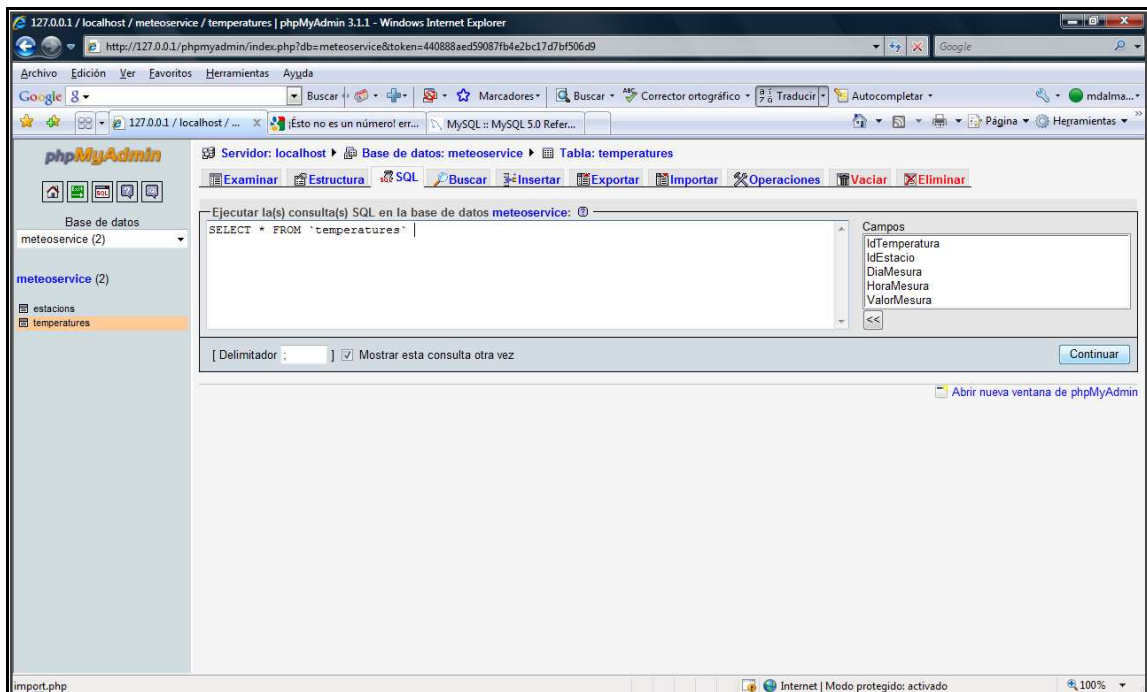




A continuació, creem la taula **temperatures**, tal i com s'ha fet per la taula Estacions. En la següent figura es mostren els camps que conté aquesta nova taula:



I una consulta de registres, per mostrar tal i com queden organitzats a la taula:



Servidor: localhost Base de datos: meteoservice Tabla: temperatures

Mostrando registros 0 - 29 (2.520 total. La consulta tardó 0.0006 seg)

```
SELECT *
FROM 'temperatures'
LIMIT 0, 30
```

Mostrar: 30 filas empezando de 30 en modo horizontal y repetir los encabezados cada 100 celdas

Número de página: 1

Organizar según la clave: Ninguna

	IdTemperatura	IdEstacio	DiaMesura	HoraMesura	ValorMesura
<input type="checkbox"/>	1	1	2009-01-01	00:00:00	-2
<input type="checkbox"/>	2	1	2009-01-01	01:00:00	-1
<input type="checkbox"/>	3	1	2009-01-01	02:00:00	-1
<input type="checkbox"/>	4	1	2009-01-01	03:00:00	-1
<input type="checkbox"/>	5	1	2009-01-01	04:00:00	0
<input type="checkbox"/>	6	1	2009-01-01	05:00:00	0
<input type="checkbox"/>	7	1	2009-01-01	06:00:00	-0
<input type="checkbox"/>	8	1	2009-01-01	07:00:00	1
<input type="checkbox"/>	9	1	2009-01-01	08:00:00	1
<input type="checkbox"/>	10	1	2009-01-01	09:00:00	2
<input type="checkbox"/>	11	1	2009-01-01	10:00:00	3
<input type="checkbox"/>	12	1	2009-01-01	11:00:00	4
<input type="checkbox"/>	13	1	2009-01-01	12:00:00	4
<input type="checkbox"/>	14	1	2009-01-01	13:00:00	5
<input type="checkbox"/>	15	1	2009-01-01	14:00:00	5
<input type="checkbox"/>	16	1	2009-01-01	15:00:00	6
<input type="checkbox"/>	17	1	2009-01-01	16:00:00	5
<input type="checkbox"/>	18	1	2009-01-01	17:00:00	5
<input type="checkbox"/>	19	1	2009-01-01	18:00:00	4
<input type="checkbox"/>	20	1	2009-01-01	19:00:00	4
<input type="checkbox"/>	21	1	2009-01-01	20:00:00	3
<input type="checkbox"/>	22	1	2009-01-01	21:00:00	3
<input type="checkbox"/>	23	1	2009-01-01	22:00:00	2
<input type="checkbox"/>	24	1	2009-01-01	23:00:00	1
<input type="checkbox"/>	25	1	2009-01-02	00:00:00	1
<input type="checkbox"/>	26	1	2009-01-02	01:00:00	-1
<input type="checkbox"/>	27	1	2009-01-02	02:00:00	-1
<input type="checkbox"/>	28	1	2009-01-02	03:00:00	0
<input type="checkbox"/>	29	1	2009-01-02	04:00:00	-1
<input type="checkbox"/>	30	1	2009-01-02	05:00:00	0

Operaciones sobre los resultados de la consulta:

Vista de impresión Previsualización para imprimir (documento completo) Exportar CREATE VIEW

Finalment, creem les altres dues taules per tenir completa la Base de Dades de MeteoService:

127.0.0.1 / localhost / meteoservice | phpMyAdmin 3.1.1 - Windows Internet Explorer

http://127.0.0.1/phpmyadmin/index.php?db=meteoservice&token=440888aed39087fb4e2bc17d7bf506d9

Base de datos: meteoservice (4)

meteoservice (4)

- estacions
- humitatrelativa
- precipitacions
- temperatures

Servidor: localhost Base de datos: meteoservice

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Eliminar

Tabla	Acción	Registros <sup>1</sup>	Tipo	Cotejamiento	Tamaño	Residuo a depurar	
estacions	<input type="checkbox"/>	3	MyISAM	latin1_swedish_ci	2.1 KB	-	
humitatrelativa	<input type="checkbox"/>	24	MyISAM	latin1_swedish_ci	2.4 KB	-	
precipitacions	<input type="checkbox"/>	24	MyISAM	latin1_swedish_ci	2.4 KB	-	
temperatures	<input type="checkbox"/>	2,520	MyISAM	latin1_swedish_ci	73.8 KB	-	
4 tabla(s)		Número de filas	2,571	MyISAM	latin1_swedish_ci	80.8 KB	0 Bytes

↑ Marcar todos/as / Desmarcar todos Para los elementos que están marcados:

Vista de impresión Diccionario de datos

Crear nueva tabla en la base de datos meteoservice

Nombre: Número de campos:

Continuar

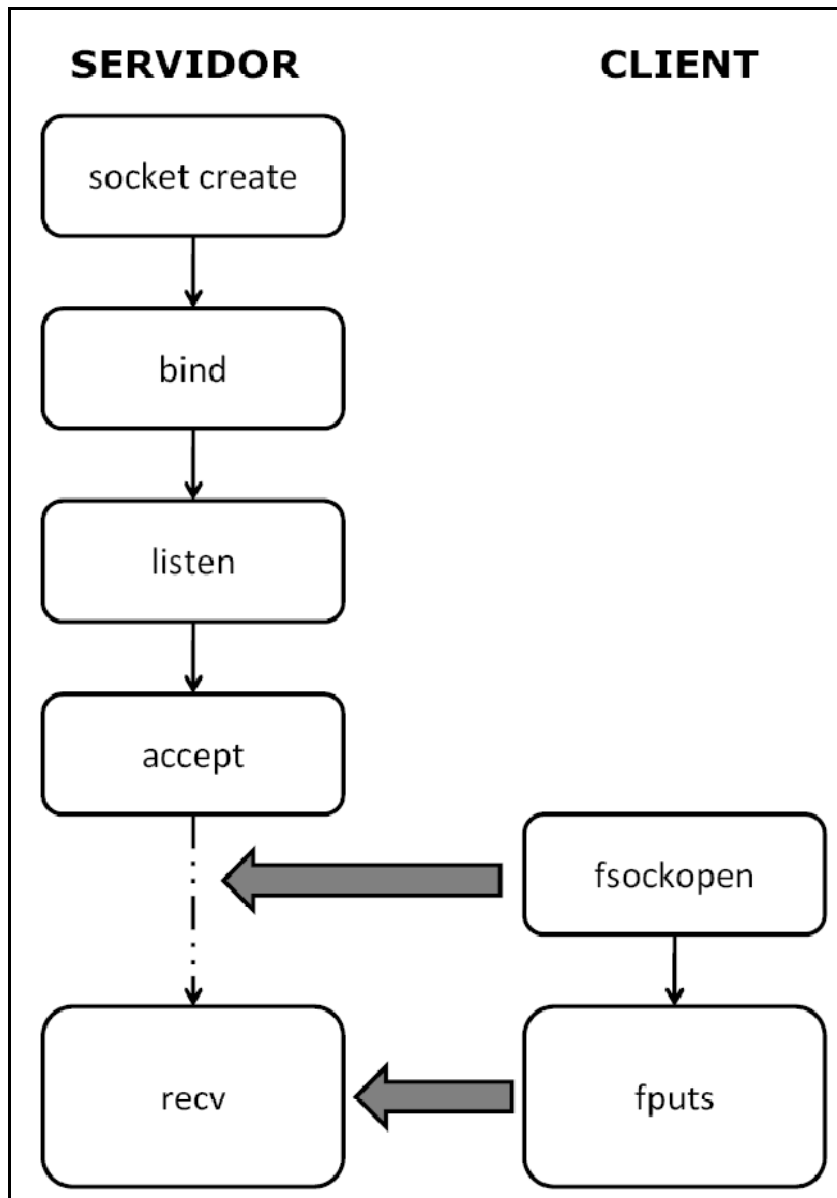
<sup>1</sup> Podría ser aproximado. Léase la FAQ 3.11

Abrir nueva ventana de phpMyAdmin

## 7.2 Descripció instruccions de sockets

El tipus de socket que utilitzem és el `SOCK_STREAM`, orientats a connexió, que permeten una comunicació bidireccional, garanteixen l'entrega segura dels paquets i la informació ordenada al destí, utilitzant el protocol TCP/IP.

En la següent figura es pot observar el flux de comunicació entre el servidor i el client de sockets i, a continuació es detallen les instruccions que empren, amb tots els paràmetres necessaris per executar-les.





Pel què fa al **servidor**:

### Socket\_create

recurs **socket\_create** (int \$domini, int \$tipus, int \$protocol)

Crea i retorna una instància de socket, també referida com a un punt final en una comunicació. La comunicació més habitual és la constituïda per dos sockets, un que realitzi el rol de client, i l'altra realitzant el rol de servidor.

Retorna la instància de socket o *false* en cas d'error.

### Socket\_bind

booleà **socket\_bind** (resource \$socket, string \$adreça, [, int \$port])

Vincula l'adreça IP donada al socket descrit per la variable \$socket. Cal fer aquesta acció abans d'establir una connexió amb `socket_listen`.

### Socket\_listen

booleà **socket\_listen** (recurs \$socket, [, int \$backlog])

Després que el socket `socket` hagi estat creat utilitzant `socket_create()` i enganxat amb el `socket_bind()`, pot començar a escoltar les connexions entrants.

### Socket\_accept

recurs **socket\_accept** (recurs \$socket)

Després d'haver executat totes les ordres anteriors, aquesta funció acceptarà connexions entrants en aquest socket. Quan una connexió ha estat exitosa, un nou recurs de socket serà retornat, i serà l'utilitzat per la comunicació.

### Socket\_recv

int **socket\_recv** ( recurs \$socket, string buf, int len, int flags )

Aquesta funció permet rebre dades del socket connectat. L'enter retornat és el número de bytes rebuts, i les dades de la trama s'extreuen de `buf`.

Pel què fa al **client**:

**fsockopen**

recurs **fsockopen** ( string destí [, int port [, int &errno [, string &errstr [, float temps\_espera]]]] )

Inicia una connexió a través de sockets amb el recurs especificat per destí.

**fputs**

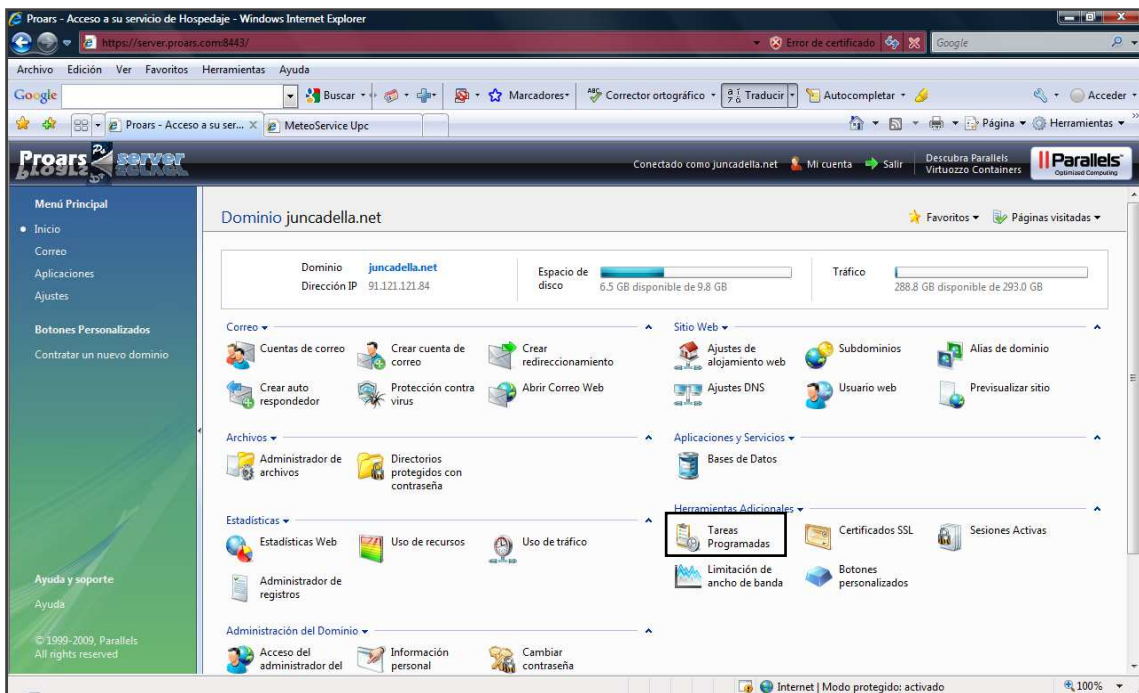
**fputs** ( recurs \$socket, string buf)


Aquesta funció permet enviar dades pel socket connectat. Les dades de la trama s'envien pel buf.

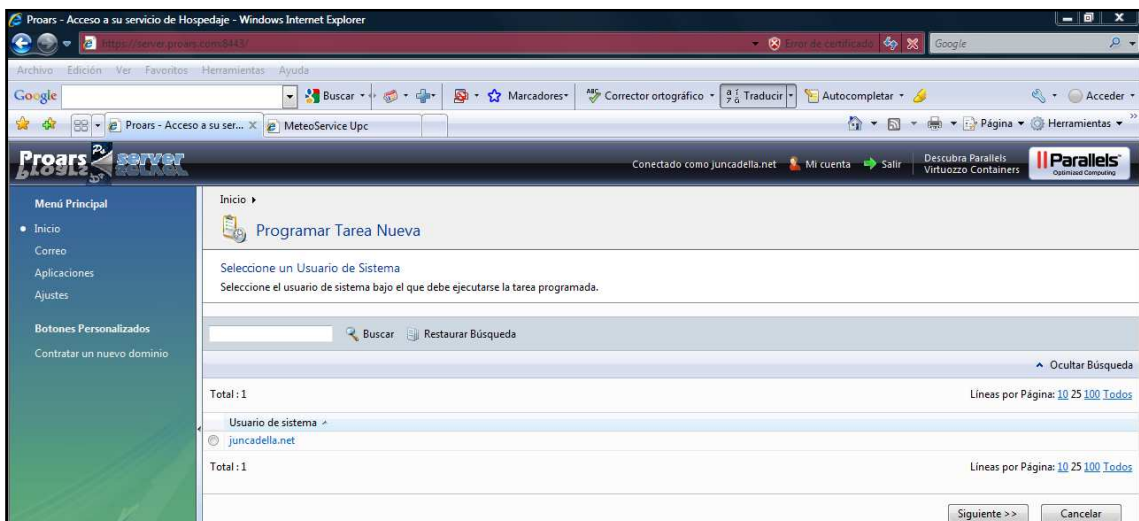
## 7.3 Programació cron job des del servidor

En el següent apartat mostrarem com es programa un cron job per fer que el servidor de sockets s'estigui executant sempre i els 3 emuladors d'estacions ho facin a cada hora (per fer de clients de sockets i enviar dades al servidor).

Des del servidor web, premem **Tareas Programadas** de l'apartat "Herramientas Adicionales".



A continuació, premem damunt de l'usuari del sistema (següent figura ) i, en la pantalla que se'ns apareix,  Programar Tarea Nueva .



Apareix un formulari en blanc per introduir els paràmetres:

Proars - Acceso a su servicio de Hospedaje - Windows Internet Explorer

https://server.proars.com/8443/

Archivo Edición Ver Favoritos Herramientas Ayuda

Google Buscar

Proars - Acceso a su servicio de Hospedaje

Conectado como juncadella.net Mi cuenta Salir Descubre Parallels Virtuozzo Containers Parallels Optimized Computing

Menú Principal

- Inicio
- Correo
- Aplicaciones
- Ajustes
- Botones Personalizados
- Contratar un nuevo dominio

Ayuda y soporte

Ayuda

© 1999-2009, Parallels All rights reserved

Inicio Usuarios de Sistema Tareas Programadas

Editar la tarea Crontab de juncadella.net

Favoritos Páginas visitadas Nivel Superior

Tarea

Activado ☒

Minuto \*

Hora \*

Día del mes \*

Mes \*

Introduzca un valor en formato crontab de Unix, por ejemplo 1, 5-7, \*/4

Seleccione un mes Enero

Día de la semana \*

Introduzca un valor en formato crontab de Unix, por ejemplo 1, 4-6

Seleccione un día de la semana Domingo

Comando \*

\* Campos obligatorios

Aceptar Cancelar

Internet | Modo protegido: activado

En la següent figura es mostra com s'ha programat el servidor de sockets (server.php) per tal que s'executi:

Proars - Acceso a su servicio de Hospedaje - Windows Internet Explorer

https://server.proars.com/8443/

Archivo Edición Ver Favoritos Herramientas Ayuda

Google Buscar

Proars - Acceso a su servicio de Hospedaje

Conectado como juncadella.net Mi cuenta Salir Descubre Parallels Virtuozzo Containers Parallels Optimized Computing

Menú Principal

- Inicio
- Correo
- Aplicaciones
- Ajustes
- Botones Personalizados
- Contratar un nuevo dominio

Ayuda y soporte

Ayuda

© 1999-2009, Parallels All rights reserved

Inicio Usuarios de Sistema Tareas Programadas

Editar la tarea Crontab de juncadella.net

Favoritos Páginas visitadas Nivel Superior

Tarea

Activado ☒

Minuto \* 0

Hora \* \*

Día del mes \*

Mes \*

Introduzca un valor en formato crontab de Unix, por ejemplo 1, 5-7, \*/4

Seleccione un mes Enero

Día de la semana \*

Introduzca un valor en formato crontab de Unix, por ejemplo 1, 4-6

Seleccione un día de la semana Domingo

Comando \*/usr/bin/php /var/www/vhosts/juncadella.net/httpdocs/meteoservice/

\* Campos obligatorios

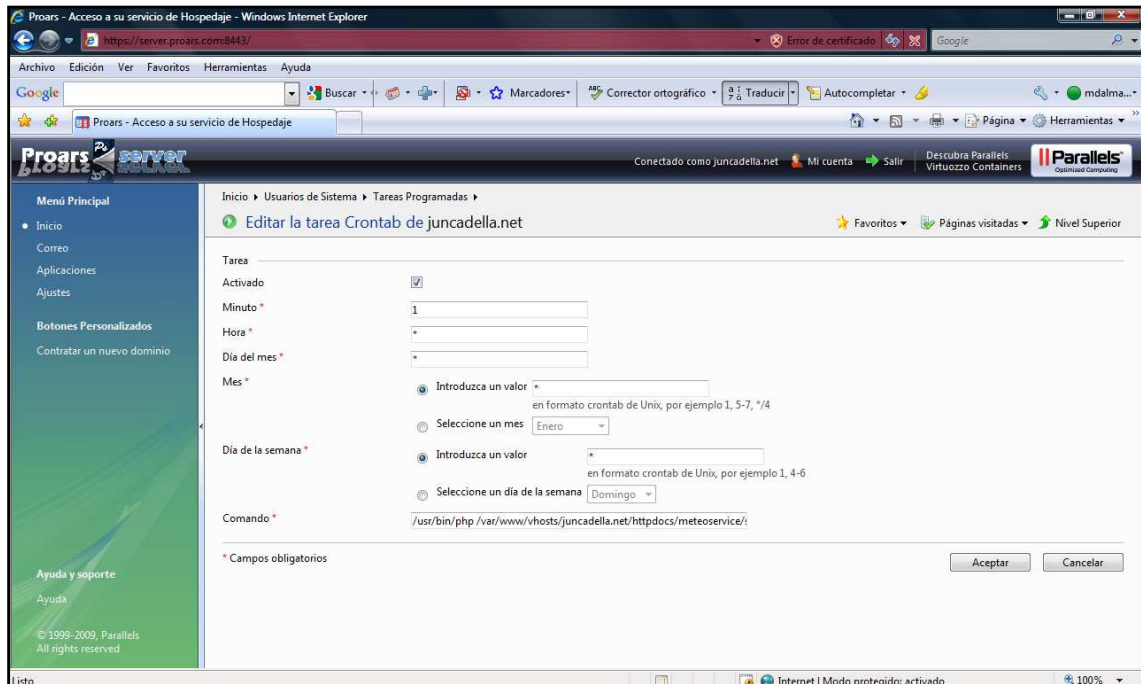
Aceptar Cancelar

Internet | Modo protegido: activado

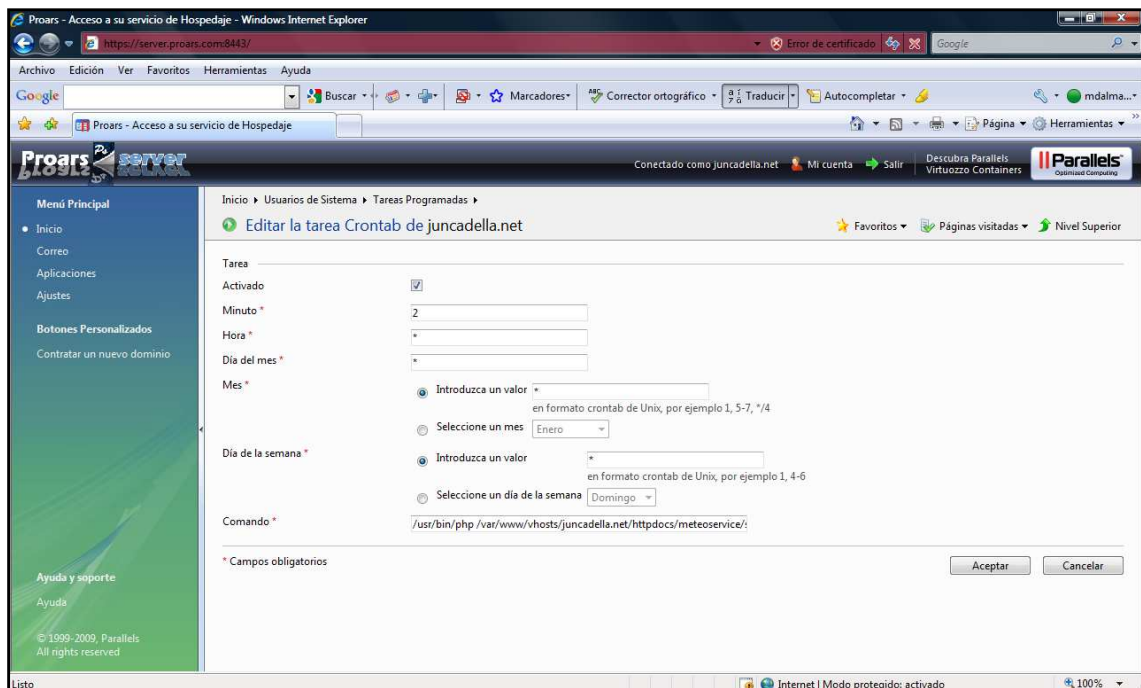
On podem observar que es que es marca que s'activi el minut 0 de cada hora, de cada dia del mes, de cada mes i de cada dia de la setmana. Hem parametritzat el servidor d'aquesta manera perquè s'executi a cada hora en punt, per esperar les

peticions de les estacions; mitjançant la funció `set_time_limit(0)`, fem que el `server.php` corri sense límit de temps.

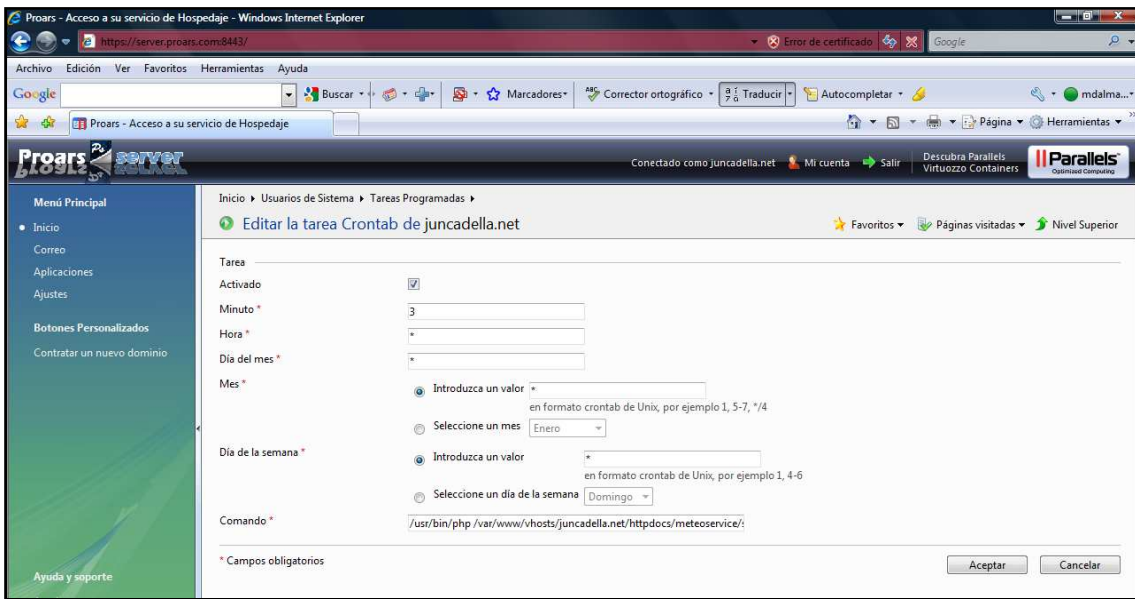
En les tres figures següents es mostren els cron de les estacions:



On es parametriza que el fitxer `estacio1.php` s'executi cada minut 1 de cada hora, de cada dia del mes, de cada mes i de cada dia de la setmana.

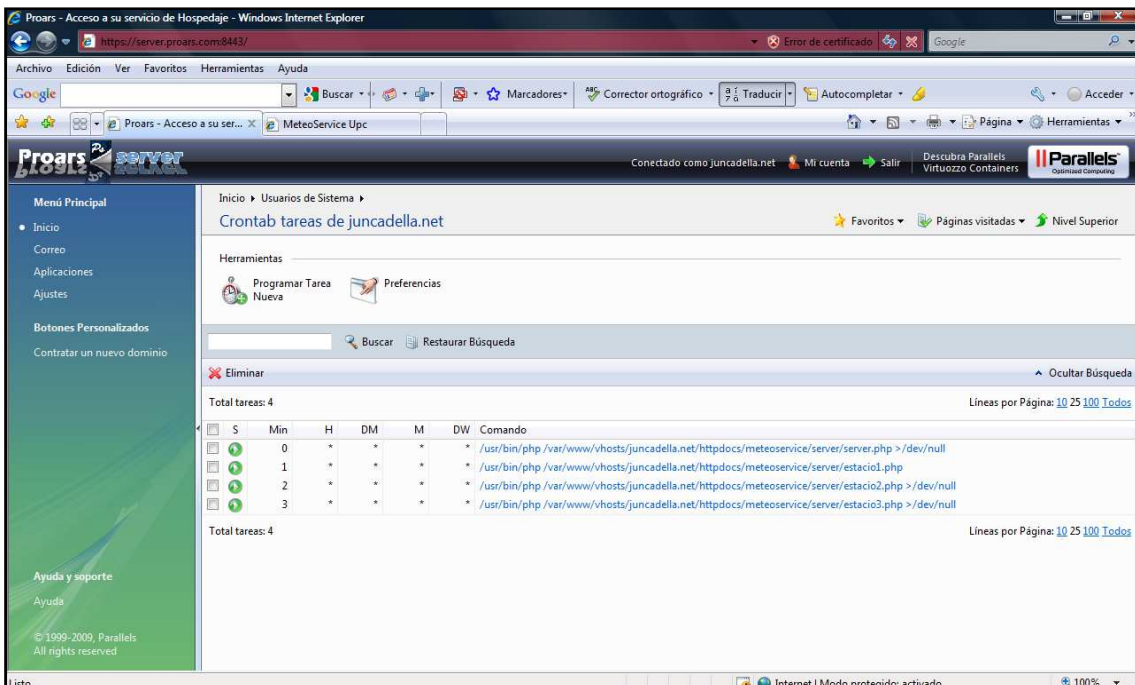


On es parametriza que el fitxer `estacio2.php` s'executi cada minut 2 de cada hora, de cada dia del mes, de cada mes i de cada dia de la setmana.



On es parametrizta que el fitxer estacio3.php s'executi cada minut 3 de cada hora, de cada dia del mes, de cada mes i de cada dia de la setmana.

Les instruccions (Comandos) passats per paràmetre, es poden veure de forma completa en la següent figura:



```
/usr/bin/php/var/www/vhosts/juncadella.net/httpdocs/meteoservice/server/server.php
/usr/bin/php/var/www/vhosts/juncadella.net/httpdocs/meteoservice/server/estacio1.php
/usr/bin/php/var/www/vhosts/juncadella.net/httpdocs/meteoservice/server/estacio2.php
/usr/bin/php/var/www/vhosts/juncadella.net/httpdocs/meteoservice/server/estacio3.php
```

La primera part de la línia de comanda indica el path de l'íntèrpret pel codi php que segueix a continuació; en el nostre cas, el binari php resideix a `/usr/bin`.

I pel què fa a la segona part, és el path on l'íntèrpret ha d'anar a buscar el fitxer php que ha d'executar (`/var/www/vhosts/domini.es/httpdocs/nom.php`).

# 8 Bibliografia

---

- **Programación de Aplicaciones Web** (Ed. Thomson)
- **Diseño Web – Guía de Referencia** (Ed. Anaya Multimedia)
- **Comunicaciones y Redes de Computadores** (Ed. Pearson Prentice Hall)
- **Introducción al SQL para usuarios y programadores** (Ed. Thomson)
- **MySQL para Windows y Linux** (RA-MA Editorial)

## Enllaços d'interès:

- [http://www.programacion-ard.com/Macromedia\\_Dreamweaver.php](http://www.programacion-ard.com/Macromedia_Dreamweaver.php)
- <http://www.manualdephp.com/>
- [http://livedocs.adobe.com/dreamweaver/8\\_es/using/wwhelp/wwhimpl/js/html/wwhelp.htm](http://livedocs.adobe.com/dreamweaver/8_es/using/wwhelp/wwhimpl/js/html/wwhelp.htm)
- <http://es.php.net/>
- [http://www.webtaller.com/construccion/programas/editores/dreamweaver/lecciones/formularios\\_dreamweaver.php](http://www.webtaller.com/construccion/programas/editores/dreamweaver/lecciones/formularios_dreamweaver.php)
- <http://www.webestilo.com/php/>
- <http://www.lawebdelprogramador.com/>
- <http://www.superhosting.cl/manuales/manual-tutorial-de-php.html>
- <http://www.desarrolloweb.com/manuales/12/>
- <http://www.google.es>